ACTIVE STEREO VISION FOR PRECISE AUTONOMOUS VEHICLE HITCHING

by

Michael Feller

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science in Mechanical Engineering

School of Mechanical Engineering

West Lafayette, Indiana

December 2019

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Song Zhang, Chair

School of Mechanical Engineering

Dr. Peter Meckl

School of Mechanical Engineering

Dr. George Chiu

School of Mechanical Engineering

Approved by:

Dr. Nicole Key

Head of the Graduate Program

This work is dedicated to Dad, Grandpa, Nick, Dave, and Dan, five men who taught me the things that college could not – to follow Jesus with my life.

ACKNOWLEDGMENTS

This work could not have been completed without help from many people, directly and indirectly. I am very grateful to Dr. Meckl and Dr. Chiu, for serving on my advisory committee and taking the time to review my thesis and defense. I am especially indebted to Dr. Zhang for his consistent feedback and guidance, as well as his willingness to allow me to work in his lab for the past couple of years.

Wissam El-Ratal was instrumental in the early phases of this project, providing motivation for the work and helping with defining the problem and need for this project.

All of my colleagues in the lab past and present have been a great help to me and I am thankful for their friendship. Many of them were helpful in providing feedback in my research and for my defense presentation. Many thanks to Michael, Xiaowei, Jialei, Saw Yan, Dr. Zhong, Dr. Marrugo, Dr. Romero, George, Yatong, Tyler, Peter, Chufan, and Ziping. In particular, I want to thank Yi-Hong and Brian for their help on my project, as they worked closely with me on various portions of the project and helped me with experiments, writing and organizing code, and hardware design. Last but certainly not least, I want to thank Jae-Sang for his immense help in the project, as he not only did preliminary work on this project that was helpful to me, but also provided consistent technical advice and valuable guidance on countless areas of the project.

I am also thankful for my friends and mentors in Purdue Bible Fellowship, at Faith West, and back home who have supported and encouraged me consistently.

I am very thankful for my family, immediate and extended, especially Dad, Mom, Olivia, Ashlyn, Grandpa and Grandma Schieler, and Grandma Feller, because without their support and love I would never have been able to complete a college degree.

Galatians 1:3-5.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Author: Feller, Michael. MSME
Institution: Purdue University
Degree Received: December 2019
Title: Active Stereo Vision for Precise Autonomous Vehicle Hitching
Committee Chair: Dr. Song Zhang

This thesis describes the development of a low-cost, low-power, accurate sensor designed for precise, feedback control of an autonomous vehicle to a hitch. Few studies have been completed on the hitching problem, yet it is an important challenge to be solved for vehicles in the agricultural and transportation industries. Existing sensor solutions are high cost, high power, and require modification to the hitch in order to work. Other potential sensor solutions such as LiDAR and Digital Fringe Projection suffer from these same fundamental problems.

The solution that has been developed uses an active stereo vision system, combining classical stereo vision with a laser speckle projection system, which solves the correspondence problem experienced by classic stereo vision sensors. A third camera is added to the sensor for texture mapping. As a whole, the system cost is $188, with a power usage of 2.3 W.

To test the system, a model test of the hitching problem was developed using an RC car and a target to represent a hitch. In the application, both the stereo system and the texture camera are used for measurement of the hitch, and a control system is implemented to precisely control the vehicle to the hitch. The system can successfully control the vehicle from within 35º of perpendicular to the hitch, to a final position with an overall standard deviation of 3.0 mm of lateral error and 1.5º of angular error. Ultimately, this is believed to be the first low power, low cost hitching system that does not require modification of the hitch in order to sense it.

1. INTRODUCTION

1.1 Autonomous Vehicles

In recent years, advancements in the development of autonomous vehicles have been a major area of study, with breakthroughs occurring regularly in many different industries. In the early 2000s, DARPA challenges allowed a variety of groups to compete in autonomous vehicle development and competitions, increasing excitement at the possibility of achievable vehicle autonomy. Recently, many major automotive manufacturers have released production vehicles with active safety features such as autonomous lane centering, active braking, and adaptive cruise control, and there is an expectation that new innovations will be developed and released rapidly in upcoming years. In the agricultural industry, self-driving tractors have been commercially available for a few years. Furthermore, there is significant potential that “swarm farming” could soon transform the industry, with many small, autonomous agricultural vehicles that could operate more efficiently compared to a single, larger vehicle. Development is underway in other industries too, and the technology is likely to soon transform vehicles in the construction and mining fields. In order to enable all of these advancements, sensing methods have been explored and applied to these technologies.

1.2 The Hitching Problem

One particular problem is very important for the advancement of autonomous vehicles is the hitching problem. The hitching problem is a complex problem, which requires sensing of a hitch and then controlling the vehicle to attach to that hitch. This problem is different from many other problems in autonomous vehicles, as precise, centimeter-level accuracy movement is crucial to successfully complete a hitching maneuver, whereas in maneuvers such as lane centering or harvesting in agriculture, for example, a lower threshold of accuracy is sufficient. However, it is an important problem, as the hitching process can be time-consuming and in some cases unsafe. As an additional challenge in many applications, the sensor used for autonomous hitching must be able to work in close proximity with other sensors of the same type, for example in applying the system to “swarm farming.”

While hitching is a significant problem to solve in the autonomous vehicle industry, surprisingly little academic research has been published presenting possible solutions to the problem. The primary study of the hitching problem to date used a laser range finder (using the same time-of-flight technology as LiDAR, discussed in Section 1.3.1) to control a tractor to autonomously hitch to an implement [1]. In this study, the researchers were able to control an autonomous tractor to the implement with a lateral error of 3 cm and an angular error of 2º. The laser range finder works by placing a reflective, artificial landmark of about 30×30 cm in size on the implement. Two artificial landmarks on the implement were required to achieve a hitching success rate of greater than 50%. Since artificial reflectors are required for the technology to work, significant modifications would need to be made to implements for tractors to detect them. Furthermore, the sensing components used in this study are listed with model numbers, and upon investigation of the component spec sheets, it can be discovered that the system would cost thousands of dollars and use about 30 watts of power at room temperature and over 150 watts in cold weather.

While there have been few academic studies of hitching, it is still a crucial problem that needs to be solved, and the desire for this technology is evidenced by several patents for automatic hitching that have been filed by a variety of vehicle manufacturers and suppliers. Since these patents do not correspond to technology that is currently in production, it is reasonable to assume that new technology and developments are still needed and desired in the industry. For example, General Motors has received a patent for a vehicle to autonomously sense a trailer and move to hitch onto it [2]. In a very similar patent, Ford was granted a patent to autonomously sense a hitch ball and move the vehicle to align with it, using a single camera or multiple cameras as a sensor [3]. Furthermore, John Deere has a similar patent for tractor hitching, describing the sensor used as a single camera sensor using image processing operations [4]. None of these patents provide details on the sensing mechanisms that would be required to realize the technologies proposed.

As has been established to the best knowledge of the present author, there are currently no published or in production methods for autonomously hitching a vehicle that (a) do not require modification of the implement being hitched to, (b) incorporate sensors costing in total under $1000, and (c) use less than 20 watts of power. Furthermore, the state-of-the-art lacks consistency, and more robust development could improve upon the accuracy and consistency of present methods. Additionally, it is important to use a sensing method in which multiple vehicles using the same sensing method can operate in the same space without loss of performance, which will broaden the hitching applications the sensor can be applied to. Section 1.3 will explore sensing methods that have been developed and will explore the advantages and disadvantages of using each of these methods to solve the hitching problem.

1.3 Sensing Methods

Various methods of sensing have been studied deeply, and in many cases specifically for the application to vehicle autonomy. Some of the most studied methods include LiDAR, radar, GPS, and cameras, and these methods have been effective for control of vehicles in various environments. The upcoming sections will discuss these methodologies and their benefits and drawbacks, specifically as it relates to solving the autonomous hitching problem.

1.3.1 LiDAR

The success of Light Detection and Ranging (LiDAR) as a sensor on autonomous vehicles has been demonstrated repeatedly [5] [6] [7]. LiDAR uses rapidly emitted laser pulses to recover a point cloud of a scene using the time-of-flight method, and can reconstruct a scene up to distances on the order of 100m. LiDAR traditionally has a cost of tens of thousands of dollars, although recent efforts have made more affordable. LiDAR has been successful at determining object locations and tracking those objects, and even 16-beam LiDAR has been shown to successfully track and identify objects, and at a reduced cost of the more robust 64-beam LiDAR [8]. Advancements in the technology have resulted in large price reductions, but even the newest, most inexpensive solutions considered robust enough for production still cost hundreds of dollars [9]. One drawback of LiDAR is the lack of data density compared to that provided by cameras, especially for low cost and low power models. This is an issue when, for the hitching application, dense reconstruction is helpful for finding not only the hitch location, but also its angle relative to the sensor. Furthermore, this causes a challenge in object identification, and a camera may be required to work in tandem with the LiDAR sensor in order to identify the hitch that is intended to be coupled with. LiDAR has proven very useful in autonomous vehicles, but it is not ideal for precise navigation required for hitching, especially if a low cost and power system is desired.

1.3.2 Radar

Radar has also been incorporated into autonomous vehicle designs [7]. Radar uses the time-of-flight method, similar to LiDAR, but with radio waves, and can have very low power requirements. One advantage of radar is the ability to not only measure distance of an object, but also directly measure its speed. Typically, radar works at longer ranges that LiDAR. However, while these advantages are important in solving some problems, they are not particularly useful in hitching applications. Radar has significantly less precision and data density compared to LiDAR, making object identification and precise measurements practically impossible without additional sensing methods. Consequently, Radar is not a viable sensor for the hitching problem.

1.3.3 GPS

The Global Positioning System (GPS) is very important in autonomous vehicle development, particularly for agricultural vehicles. For any space of known configuration, such as a field or highway, GPS can be used to locate that position and provide feedback. One problem, however, is that present GPS methods have errors on the order of meters, much too large for precise navigation in a hitching application [10]. However, even with the possibility of improvements to the accuracy on the order of centimeters, in hitching the position and orientation of the hitch is not assumed to be exactly known, so using GPS alone is not a viable solution for completing a hitching maneuver, as some feedback sensing of the hitch is needed.

1.3.4 Cameras

In autonomous vehicle development, cameras have been perhaps the most important sensor due to the density of information they can provide. A wide range of camera-based sensors have been developed, many of which have been used on autonomous vehicles. Single cameras have been applied as a sensor in various ways. Multiple cameras have been used for 3D reconstruction in stereo vision. Structured light techniques have incorporated cameras in coordination with a projector to obtain a 3D reconstruction. Infrared cameras, event cameras, the emergence of artificial intelligence, and other technologies have opened up many related, nuanced areas of research in this field.

1.3.4.1 Time-of-Flight

Cameras have been used successfully for the time-of-flight method [11]. For example, an infrared camera has been used to measure the time between infrared light emission and detection, resulting in centimeter level uncertainty at short distances using about 10 W of power [12]. Additionally, the commercially available Microsoft Kinect V2 system uses the time-of-flight method, and consumes slightly more power than the aforementioned system. The system is also not recommended for close range use at distances around or under 1m, which poses some problems for hitching in close proximity. For hitching applications, the accuracy limitations of the time-of-flight method could pose a problem in creating a reliable signal for vehicle control.

1.3.4.2 Structured Light

Structured light methods have been studied, and many variations exist [13]. In principle, structured light systems use some projection system to project a known light pattern onto a scene, and then a camera captures the scene. Random patterns have been used in projection in several commercial sensors, including the Microsoft Kinect V1 and the Intel RealSense R200. If the pattern is known, the results can be projected to 3D using triangulation. As an advantage, the technology can be inexpensive and small enough to be used in mobile applications. However, of special interest in the hitching application, the spatial resolution of the data is low, due to limited capacity for projected feature density, which can make object orientation measurement challenging. Furthermore, there are limitations to the measurement accuracy, which in combination to the low feature density can limit the capability of these sensors for precise feature measurement. Furthermore, the Kinect V1 system is not recommended for use in the range of around or less than 1 m, which poses problems for the hitching application.

Digital fringe projection (DFP) systems have been studied extensively as another structured light system, and have shown great progress for high accuracy and high speed 3D measurement [14] [15]. In this method, the projection contains an intensity pattern that varies sinusoidally, and the phase of the intensity is able to be computed and used for shape measurement. High accuracy measurement has been demonstrated, and some systems achieve this high accuracy in the kHz range, showing the capability of DFP systems to be used in a wide variety of applications. However, systems achieving these results require tens of watts of power to project high intensity light, and also require high cost projectors and cameras, resulting in a sensor with total cost in the thousands of dollars. Furthermore, structured light techniques have interference problems when multiple sensors are used in the same space, and also due to changes in ambient light. Some techniques have been attempted to solve this problem, but typically this requires synchronization between different sensors, which may not be realistic in real world solutions.

1.3.4.3 Stereo Vision

Stereo vision has been studied as well and is of significant importance in the development of computer vision [16] [17]. Stereo vision, like the human eyes, works by matching points in images corresponding to the same point in a scene. Once pixels are matched, the 3D scene can be reconstructed. Many different types of stereo matching algorithms have been developed, but mainly separate to local methods and global methods. Local methods are area-based, and use a window of pixels to find matches. Global methods are energy-based, and use the entire image, attempting to minimize a cost function. Generally speaking, global methods are very computationally expensive, and are not effective for real-time applications. Local methods can be used in real time; however, the accuracy is lower than for the global method. A further distinction of stereo matching methods is between dense and sparse reconstruction. Sparse algorithms only match small image segments or edges, which is insufficient for most modern problems, while dense algorithms, as the name implies, are capable of matching many more points in the images, and these algorithms are preferred for most applications. OpenCV has an algorithm for dense, area-based stereo matching, the block matching (BM) algorithm [18]. Another popular open-source, dense algorithm for real time stereo vision is the Efficient Large-Scale Stereo Matching (ELAS) algorithm [19]. While both of these algorithms are considered to be sufficient for real-time applications, the BM algorithm has been shown to be more efficient, while also maintaining sufficient accuracy for many real-time applications even without high computational resources [20]. As an advantage, stereo vision setups can be implemented without expensive hardware, as camera costs have dropped as the technology improves. Furthermore, many cameras have low power supply requirements, on the order of single watts or less. Of particular advantage in the hitching problem is an increase in accuracy at short distances in stereo vision. In general, stereo vision accuracy increases with the square as distance decreases, which can result in very high accuracy at short distances, at the cost of compromised accuracy at far distances. One significant problem in real world contexts for stereo matching is the problem of finding stereo correspondences [21]. In particular, objects with homogeneous texture and color content are challenging to match properly, as area-based methods cannot easily match these areas of the image. This problem is intensified for algorithms that are optimized for speed and use limited computational power.

To solve the stereo correspondence problem, the area of active stereo vision has been explored. In active stereo vision, a projection system illuminates the surface of a scene with a pattern, adding distinct features to the scene that aid the stereo algorithm in finding correspondences in the scene. For example, scene reconstructed has been completed with the simple pattern of vertical black and white stripes projected with a standard projector [22]. A colored stripe, rainbow-like projection has also been attempted. This methodology improved results and helped solve the correspondence problem [23] [24]. However, methods like these with standard projectors, such as DLP projectors, suffer from a few fundamental problems. When multiple projectors are used in the same space, the projected light can blend and lose the unique features, and therefore failing to solve the correspondence problem by projecting features onto homogeneous regions of the scene. Furthermore, these projectors frequently require a significant amount of power, an order of magnitude greater than the cameras need. In addition, projector cost can be a factor, and the size can be larger than desired for mobile applications. However, another type of projection system has been proposed that can address all of these problems experienced by traditional projectors: a laser speckle projection [25]. By projecting an objective speckle pattern using a laser system, distinct correspondences can be found in an image pair, allowing accurate 3D reconstruction. Another study has used an RGB laser beam to create the speckles for increased accuracy [26]. A laser speckle projection system has recently been applied successfully for measurements in medical applications [27]. The laser system can be inexpensive, compact, lightweight, and require well under a watt of power. Consequently, using an active stereo vision system with a laser speckle projector could be a successful sensor solution to the hitching problem.

2. ACTIVE STEREO SENSOR DEVELOPMENT

2.1 Introduction

3D optical sensing is very important in the advancement of autonomous vehicles for agriculture, transportation, construction, and a variety of other fields. Precise vehicle movement requires accurate and robust sensing, particularly for the application of autonomous hitching. Existing 3D sensing methods have fundamental limitations preventing them from being viable solutions to the hitching problem, such as high power consumption, high cost, insufficient accuracy and resolution of data, or inability to have multiple systems sensing the same area simultaneously. In order to precisely control a vehicle, a sensor will need to be developed that can solve these issues. The active stereo vision sensor developed for this project consists of two cameras combined with a laser speckle projection system. This chapter will explain the principle of the laser speckle system and stereo vision algorithms, describe the hardware and setup associated with the stereo vision system, and present experiments evaluating the accuracy and viability of the system to be used as a sensor. Furthermore, the addition of a third camera will be discussed in order to map the texture content to the 3D geometry.

2.2 Laser Speckle System

One of the major limitations of stereo vision is that many pixels are not able to be matched correctly if unique features are not present in the scene under view due to homogeneous color and texture within regions of the image [21]. This stereo correspondence problem leads to either unmatched pixels, which reduce the density of the data collected, or incorrect matching, which results in significant errors in the measurement. This section proposes a laser speckle system to solve this problem by projecting a random pattern onto the scene. The proposed system is inexpensive, light-weight, low-power, and compact. Consequently, it is a viable solution to the pixel matching limitation.

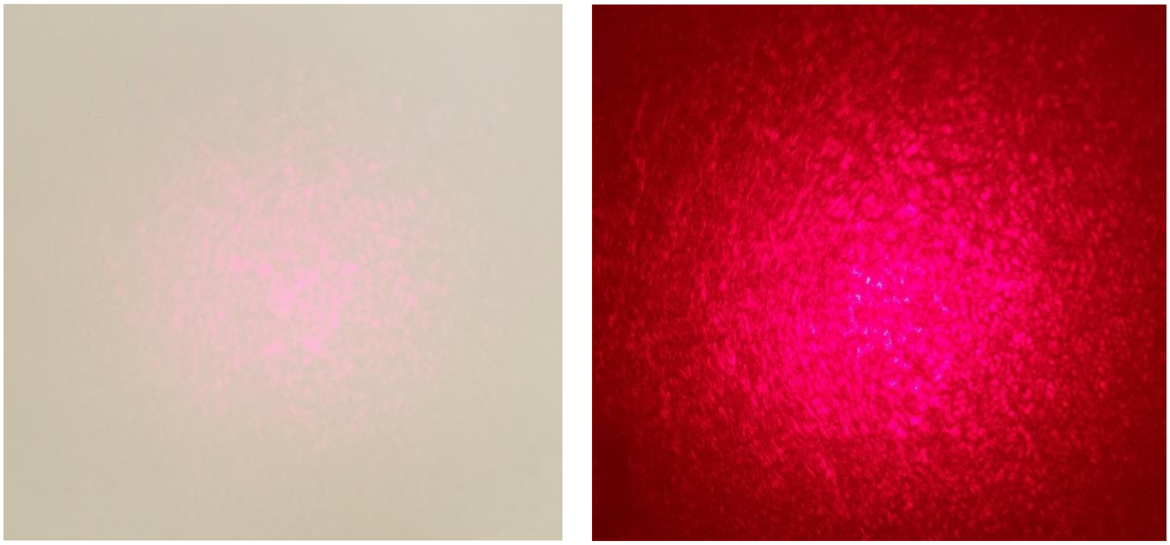
Coherent light sources, such as lasers, experience the phenomenon of interference, which is the principle that underlies the proposed projection laser speckle pattern. When a laser beam propagates through a rough surface, an objective interference pattern is generated. In practice, this can be achieved by passing the beam through a diffuser, which is an optical element that scatters light that is transmitted through it. A pattern of random speckles projected on a screen can be defined statistically by the average speckle size

$$\sigma_0 \cong \frac{1.2\lambda L}{D}, \tag{2.1}$$

where λ is the wavelength of the beam, L is the distance from the diffuser to the screen, and D is the diameter of the beam at the diffuser. The average speckle size σ is defined as the average distance between regions of maximum and minimum brightness [28]. In order to increase the size of the spread of the projection and also influence the size of the speckles created by the projection, a lens can be utilized to focus the beam onto the diffuser. The setup used to project the laser speckle pattern therefore corresponds to Figure 2.1.

Figure 2.1: Hardware setup used to project the laser speckle pattern.

In the proposed setup, an inexpensive 650nm laser diode from Lights88 was used which had a power of 50 mW and a UPC of 602105294874. In order to greatly increase the intensity of the speckle pattern while still using a low-power, compact laser, long-pass optical filters were placed on the cameras in the stereo system. The filters chosen were model number FGL645 from Thorlabs, which have a cut-on wavelength of 645 nm. This specific filter was chosen for its low cost and because it transmits at least 75% of light at wavelengths of 650 nm or longer, while transmitting less than 5% of light at wavelengths of less than 620 nm, resulting in greatly increased intensity of the speckle pattern. Figure 2.2 shows the impact of adding the optical filter to the camera.



(a) (b)

Figure 2.2: Laser speckle pattern projection onto a white wall. (a) Raw unfiltered projection; (b) Long-pass filtered image of the same projection.

With the laser speckle system able to project onto a scene, it can be used in coordination with two cameras for active stereo vision. In the setup of the system, the speckle projection system is placed between two cameras, which can capture images of the scene, and this setup is shown in Figure 2.3.

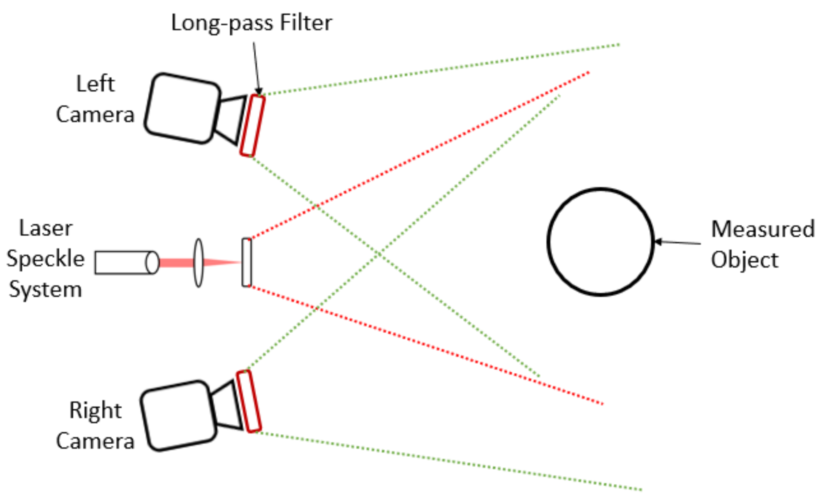


Figure 2.3: Setup of the stereo vision system cameras and laser speckle projection system.

This stereo camera system is then also shown in Figure 2.4, which shows the actual setup of the system with key components labeled, including the laser speckle projection system which sits in a housing.

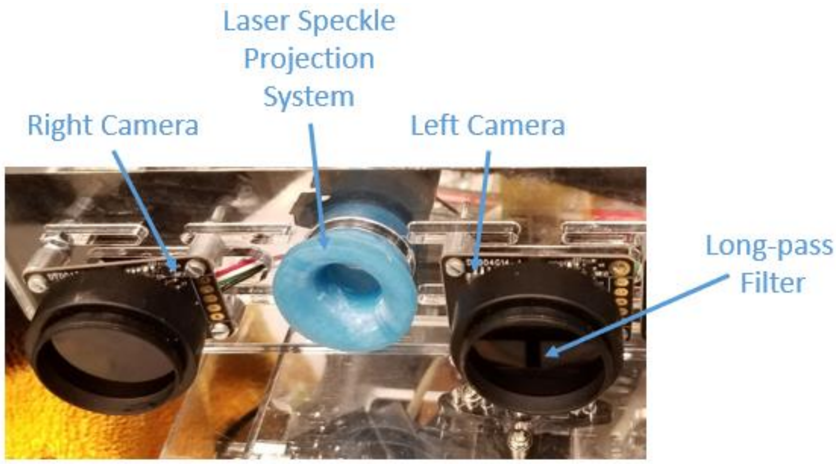


Figure 2.4: Setup of the actual stereo vision system, with key components labeled and the laser speckle projection system in a housing.

Each captured image pair from the two cameras can be used to reconstruct the scene in 3D by applying a stereo matching algorithm. Before stereo matching may be completed, however, the cameras must be mathematically modelled, and the system needs to be calibrated.

2.3 Pinhole Camera Model

Mathematically, it is important to properly and accurately model the cameras within the system. Multiple models have been proposed, however the most widely used model is the pinhole model [18]. In this model it is first important to define several coordinate systems, as shown in Figure 2.5.

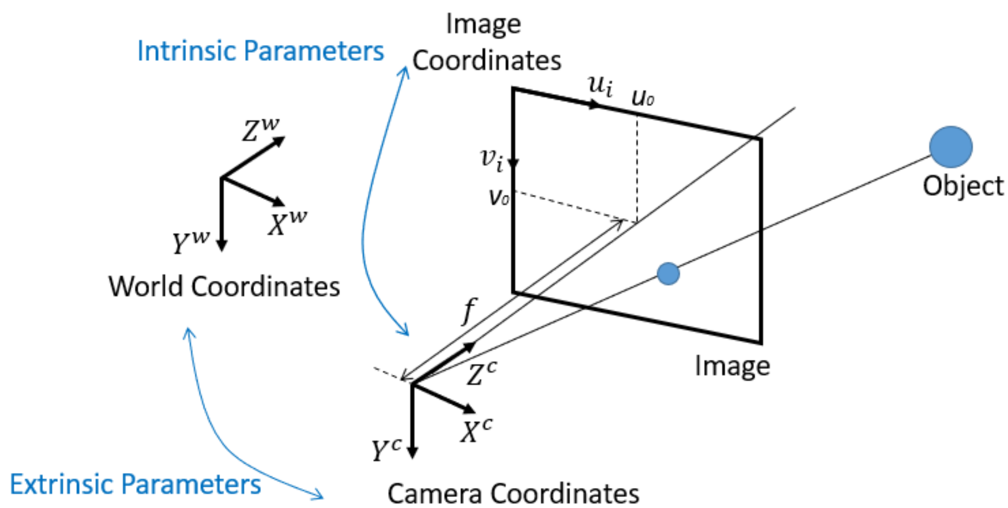


Figure 2.5: Pinhole camera model.

The overall coordinate system is the world coordinate system in which both cameras and the observed scene exist. Within that system, each camera has its own camera coordinate system, with an origin at the pinhole. The image plane sits at a distance of f, the equivalent focal length, from the camera coordinate system. The Z^c axis of the camera system is perpendicular to the image plane. Finally, the image coordinate system lies on the image plane and is defined in units of pixels. As the camera captures images of the world, the intersection of the projection line between the world point and the camera coordinate origin on the image plane creates the point on the image corresponding to that world point. The transfiguration of this geometry to the actual image is not perfect however, since the camera sensor pixels are not perfectly square. Consequently, a variety of intrinsic parameters are identified to transform to the actual image coordinates. These parameters include the focal lengths, f_x and f_y, and the principal point offset, u_0 and v_0. Note that f_x and f_y can be calculated directly during calibration, but they are the product of the physical equivalent focal length, f, and the respective pixel densities s_x and s_y, which have units of pixels per unit length. The distinction between f_x and f_y is necessary because the pixels of a real camera are not perfectly square [18]. The intrinsic parameters are encapsulated within the intrinsic camera matrix

$$\boldsymbol{A} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.2}$$

Since there are two cameras in the stereo vision system, it is also important to define extrinsic parameters of the system, which specify the camera rotation and translation from the world coordinate system. The rotation matrix $\boldsymbol{R}$ is traditionally defined as a 3×3 matrix, while the translation matrix $\boldsymbol{T}$ is represented as a 3×1 column matrix. In practice, the left camera coordinates are set as the world coordinates, resulting in an identity rotation matrix with a translation matrix of zeros. The right camera translation and rotation matrices are then given with respect to the left camera. In total, according to the pinhole camera model, a camera is mathematically described by

$$s\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \boldsymbol{A}\begin{bmatrix} X^c \\ Y^c \\ Z^c \\ 1 \end{bmatrix} = \boldsymbol{A}[\boldsymbol{R}|\boldsymbol{T}]\begin{bmatrix} X^w \\ Y^w \\ Z^w \\ 1 \end{bmatrix}, \tag{2.3}$$

where s is a scaling factor. As will be discussed in Section 2.4, the intrinsic and extrinsic parameters can be calculated for a specific camera using a camera calibration process.

2.4 Camera Calibration

In order to obtain the world coordinates of a point captured by the stereo camera system described in Section 2.3, the system must first be calibrated. Proper calibration will result in accurate calculation of the intrinsic and extrinsic camera parameters. The most common calibration method for stereo camera systems is to use a calibration board with a known pattern. For intrinsic calibration of each camera, a variety of poses of the calibration board can be captured. For high accuracy of the calibration, Zhang proposed using a circle pattern board for the calibration, and this pattern is shown in Figure 2.6 [29]. Precise identification of the circle centers can be achieved using OpenCV. Following identification of the circle centers in the image, OpenCV algorithms can be used to find all the intrinsic parameters of the camera [18]. The cameras used for this project claim to have non-distortion lenses. Experimental results confirmed that lens distortion could be ignored, as including distortion parameters does not significantly impact measurement results, which allowed for reduction of complexity during calibration and camera modeling.

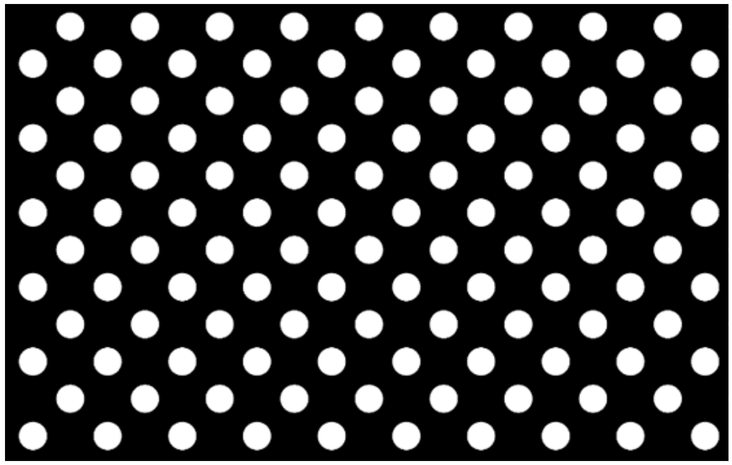


Figure 2.6: Calibration circle pattern used to calibrate the stereo camera system.

In addition to the intrinsic parameters, extrinsic parameters must be obtained which relate the camera coordinates to the world coordinates. Specifically, this requires finding the rotation and translation of the cameras with respect to the world coordinate system. Equation (2.3) shows the relationship between the coordinate systems. To simplify this analysis, one of the cameras can be set as the origin of the world coordinate system (in this project, the left camera), resulting in that camera having an identity rotation matrix and a translation matrix of zeros. By capturing various poses of the calibration board simultaneously with both cameras, the required transformations can be computed using OpenCV by matching the locations of the circles in the captured images at various poses and using the previously calculated intrinsic parameters, ultimately allowing the extrinsic parameters to be obtained [18].

2.5 Block Matching Algorithm

After calibration, a stereo matching algorithm can be utilized to obtain a disparity image from the dual camera images. The block matching algorithm was chosen as the best option for this application. The algorithm was implemented using the OpenCV StereoBM class [18].

Fundamentally, the block matching algorithm, similar to most stereo vision algorithms, requires rectified images to work properly. Visually, rectified images are images that are transformed so that correspondences in both images are on the same horizontal row of pixels. Importantly, this turns the pixel matching search into a 1D problem on a known row of pixels rather than a 2D problem requiring search of the entire image. This transformation is made possible geometrically by the existence of epipolar lines, which are shown for an object viewed by a stereo vision system in Figure 2.7. With the extrinsic and intrinsic parameters known following calibration, the image can be remapped so that the epipolar lines in both the left and right image lie on the same horizontal row of pixels. This mapping is able to be completed using the functions stereoRectify, initUndistortRectifyMap, and remap [18].

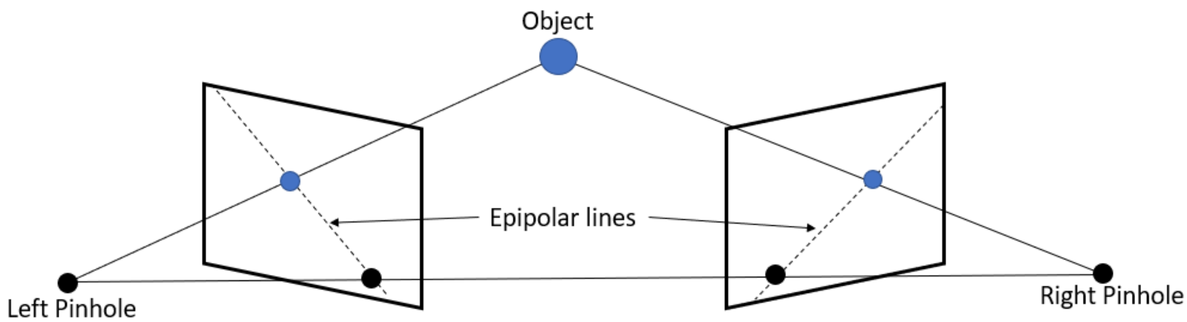


Figure 2.7: Epipolar lines for an object viewed by a stereo vision system.

Once rectification has been completed, the block matching algorithm can be used. The fundamental principle of the block matching algorithm is minimizing the sum of absolute differences between two blocks of pixels on two images. For a pixel in the rectified left image, *Lr*, the sum of absolute differences (*SAD*) is computed as

$$SAD_k(u,v) = \sum_{m=u-L}^{u+L} |Lr(m,v) - Rr(m-k,v)| \tag{2.4}$$

for a linear horizontal window of size 2*L*+1 centered around pixel (*u*, *v*) compared to a window of the same size that scans the same row *v* on the rectified right image, *Rr*, where *k* can be incremented to scan one row of *Rr*, and this process is shown in Figure 2.8. The minimum result will correspond to the matching pixel in the second image. The difference in column number of the matched pixels in the rectified images is the value of the disparity, σ. The results of this algorithm are stored in a disparity map. Small values in the disparity map correspond to larger distances, while large values correspond to small distances. With this step completed, a disparity map of integer values has been created.

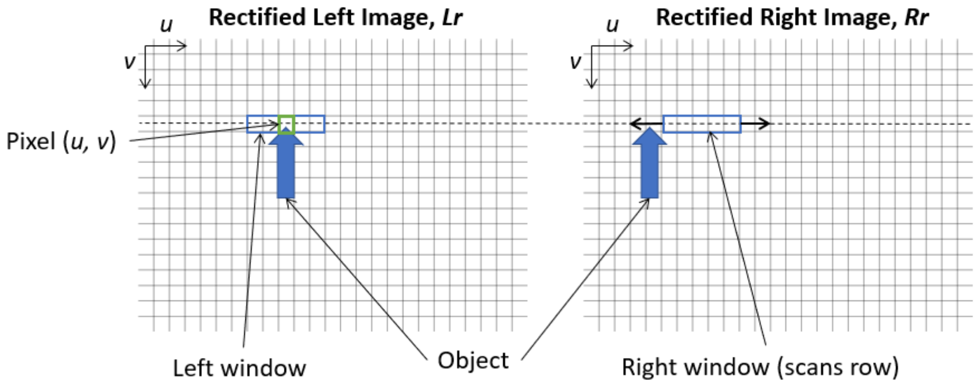


Figure 2.8: Linear window scanning the rectified right image and compared to a linear window in the rectified left image for block matching.

2.6 Sub-Pixel Matching Algorithm

Based upon what has been presented in the preceding sections, all the theory is in place to develop a typical stereo vision system and obtain a disparity map of a scene of interest. However, the results can be vastly improved by incorporating some modifications to the algorithms. A sub-pixel matching algorithm can be used to improve the disparity map. One of the most significant problems with the standard block matching algorithms is that the disparity map only includes integer disparities. The issue with this is that it results in a large depth resolution. For example, a difference in disparity of 1 pixel for an object a meter away could result in a depth difference of 1 cm (Note that this is a typical result, but the actual measurement can vary depending on the setup of the cameras, the camera resolution, and other factors). As a result, precise object reconstruction is impossible using the raw block matching algorithm. Consequently, a sub-pixel algorithm developed by McCormick has been adapted for increased accuracy [30].

The principle of this algorithm is similar to the block matching algorithm. For a given pixel at location (u, v) in the rectified left image, a square block centered on that pixel is compared with blocks centered on the same row in the rectified right image. The sums of absolute differences are computed along the row, and after scanning the minimum sum of absolute differences is then found, which corresponds to the matching pixel in the right rectified image. This should correspond to the integer disparity that the block matching algorithm would find. Using this minimum sum of absolute differences, in coordination with the sum of absolute differences of the pixel windows immediately to the left and right in the rectified right image, quadratic interpolation can be used to find the sub-pixel disparity. The three points are fit with a quadratic curve, and the minimum of that curve is the sub-pixel disparity, in accordance with

$$\delta = \sigma - \frac{\left(SAD_1(u,v) - SAD_{-1}(u,v)\right)}{2\left(SAD_{-1}(u,v) - 2SAD_0(u,v) + SAD_1(u,v)\right)}, \tag{2.5}$$

where δ is the subpixel disparity and σ is the integer disparity. $SAD_0(u,v)$, $SAD_{-1}(u,v)$, and $SAD_1(u,v)$ correspond to the sums of absolute differences of windows around the minimum pixel in the rectified right image and the pixels on its left and right, respectively, which can be computed as

$$SAD_{k-\sigma}(u,v) = \sum_{m=u-L}^{u+L} \sum_{n=v-L}^{v+L} \left|\left(Lr(m,n) - Rr(m-k,n)\right)\right|, \tag{2.6}$$

where the window of size $(2L+1)\times(2L+1)$ is centered at pixel (u, v) in the rectified left image *Lr* and compared to a window in the rectified right image *Rr*, and *k* is incremented across the row in the right rectified image. Note that Equation (2.6) essentially expands Equation (2.4) to a square window.

The algorithm proposed above has been improved in a couple of key ways. First, it was not used for every pixel in the row as proposed, as this would be too computationally expensive to be useful in the real-time application discussed in Chapter 3. Consequently, it has been adapted to be used after obtaining the disparity map from block matching, which allows the search to be narrowed. Using the known integer disparity from the block matching algorithm, the only three pixels in the rectified right image that need to be compared with the pixel of interest in the left image are the pixel corresponding exactly to the disparity found in block matching, and the pixel on its left and right, as shown in Figure 2.9. This allows the process to be computed much more quickly and allows the algorithm to be implemented strategically in real-time applications.

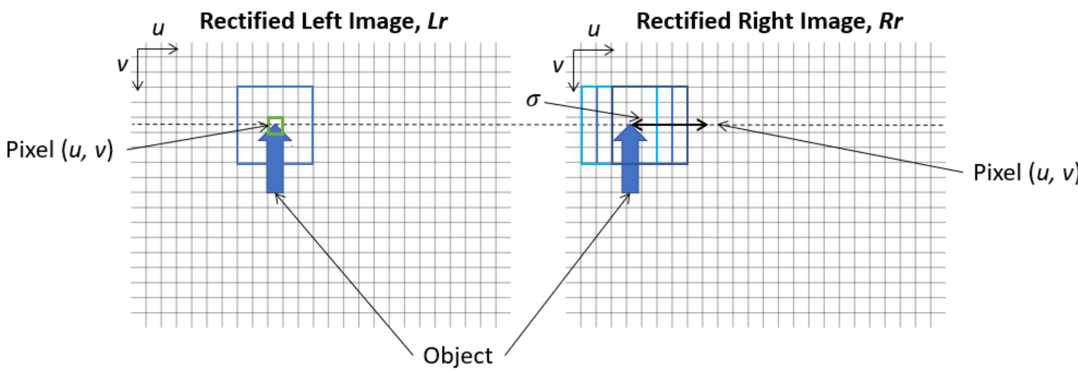


Figure 2.9: Three square windows in the rectified right image compared to a square window in the left image for sub-pixel block matching.

Another improvement made to the subpixel matching algorithm is incorporating a smooth window in the computation of the sum of absolute differences. A 2D Hann window, which is defined as

$$w(m,n) = 0.25\left(1+\cos\left(\frac{\pi m}{L}\right)\right)\left(1+\cos\left(\frac{\pi n}{L}\right)\right), \qquad (2.7)$$

where $m = -L, -L+1, \dots, 0, \dots, L-1, L$ and $n = -L, -L+1, \dots, 0, \dots, L-1, L$, is used, smoothing out some artifacts present in the sub-pixel disparity map when no window, effectively a rectangular window, was used. This window is multiplied with the square block of the sum of absolute differences between the left image area of interest and the block in the right image it is being compared to. Consequently, the three necessary sums of absolute differences can now be defined by

$$SAD_k(u,v) = \sum_{m=u-L}^{u+L}\sum_{n=v-L}^{v+L} w(m-u, n-v) * \left|\left(Lr(m,n) - Rr(m-\sigma+k, n)\right)\right|, \qquad (2.8)$$

where the window of size (2L+1)×(2L+1) is centered at pixel (u, v) of interest in the rectified left image (Lr) corresponding to the three windows in the rectified right image (Rr), where each window is determined by the parameter k being set to -1, 0, or 1. Equation (2.5) then is applied to find the subpixel disparity from quadratic interpolation.

Overall, this algorithm increased both the accuracy and visual appearance of the results, and also was instrumental in implementing the algorithm in the real-time application discussed in Chapter 3.

2.7 3D Reconstruction

3D reconstruction of the scene can be computed using the disparity values and the calibrated parameters. The equations for reconstruction are defined as

$$K\begin{bmatrix} X^w \\ Y^w \\ Z^w \\ 1 \end{bmatrix} = \boldsymbol{Q}\begin{bmatrix} u \\ v \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -u_0 \\ 0 & 1 & 0 & -v_0 \\ 0 & 0 & 0 & f \\ 0 & 0 & -\frac{1}{T_x} & \frac{u_0 - u_0'}{T_x} \end{bmatrix}\begin{bmatrix} u \\ v \\ \delta \\ 1 \end{bmatrix}, \tag{2.9}$$

where the 4×4 $\boldsymbol{Q}$ matrix consists entirely of known parameters from the calibration, u and v are the coordinates of the pixel with disparity δ [18]. Notably, from the intrinsic parameters shown in Equation (2.2), u_0 and v_0 are defined for the left camera, while u_0' is defined for the right camera. T_x is the baseline length, which is the magnitude of the right camera translation matrix $\boldsymbol{T}$. With 4 equations and 4 unknowns, the system can be solved for the world coordinates corresponding to an individual pixel in the disparity map. By applying Equation (2.9) repeatedly, the entire point cloud for the disparity map can be solved. Finally, the 3D data has been obtained. In summary, Figure 2.10 shows a summary of the processes needed to obtain 3D data from stereo vision.

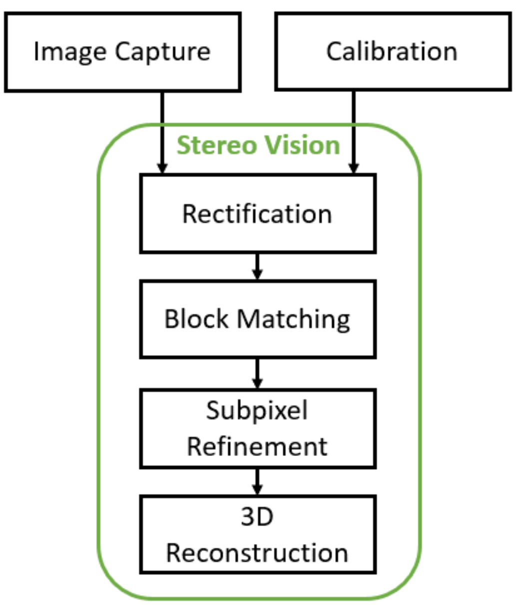


Figure 2.10: Flow diagram of the processes and algorithms used for stereo vision.

2.8 Texture Camera Incorporation

The sensor that has been developed in the preceding sections of this chapter is capable of accurate 3D reconstruction and can be used for sensing. In many contexts, however, it is important to have RGB texture data that can map to the measured 3D geometry. However, the two cameras in the stereo system have no meaningful color data, as the speckle pattern dominates the image, and most of the color content is filtered out by the optical filters. Consequently, a third camera is proposed that can map the RGB texture from the scene to the measured geometry. This camera is calibrated with respect to the left camera in the stereo vision system, as described in Section 2.4. This camera is covered with a short-pass optical filter, so the red laser light is filtered out, and as shown in Figure 2.11, which shows the entire vision system, including the stereo vision system, the laser speckle projection system, and the single texture camera.

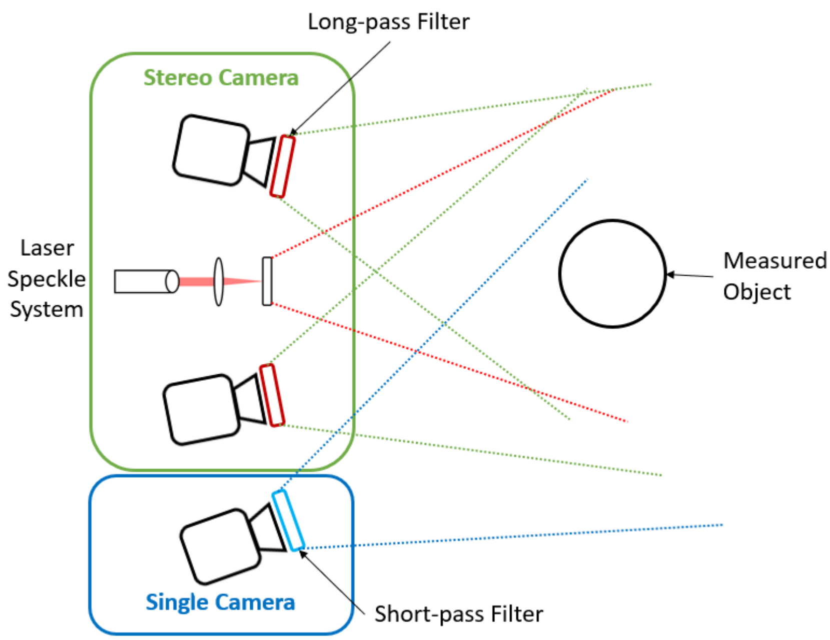


Figure 2.11: Overall sensor hardware setup including the stereo camera, the laser speckle projection system, and the single RGB camera.

Some of the red color in the scene may also be filtered out, but the color is close to the color in the real world, especially for green and blue shades. In order to map the color data with the 3D geometry, the pinhole camera model from Equation (2.3) can be used. Following block matching, each pixel in the disparity map can be projected to the world coordinates. With the world coordinates known, and all the intrinsic and extrinsic parameters known for the texture camera, Equation (2.3) can be solved for the pixel number in the texture image that corresponds to that specific world coordinate. As a result, the texture content can be accurately mapped to the 3D geometry. This method has proven to be robust for accurately mapping the color, which will be demonstrated in Section 2.13.

2.9 Hardware Overview

This section overviews the hardware used for the active stereo vision system. First, low cost cameras were chosen that have sufficient frame rate and resolution options to work effectively for the project. Each camera has a ¼ inch CMOS sensor, can capture at rates up to 30 fps, and has a variety of resolution options up to 1280×1024. The camera is USB 2.0 compatible and is powered through the USB BUS at a power consumption of 0.75 W. As an added advantage, these cameras have non-distortion lenses, allowing for some simplifications in analysis. The cameras view the scene through 645 nm long-pass light filters.

The laser speckle system in this project uses a 650 nm red light laser. The beam is passed through a glass optical lens with a focal length of 10 mm and a 600-grit ground glass diffuser. The laser projection system uses just 0.05 W of power. Consequently, the entire stereo vision system uses under 3 W of power. Overall, the components used for the sensor are described in detail in Table 2.1, including the cost and power usage of each component.

Table 2.1: List of components used for the active stereo vision sensor, including their cost and power usage.

Component	Product Number	Quantity	Unit Cost ($)	Power (W)
USB Mini Camera Module	ASIN: B07CHVYTGD	3	20	0.75
Lights88 50mW Laser Diode	UPC: 602105294874	1	6	0.05
10mm Dia. x 10mm FL Lens	Edmund Optics Stock#: 63-471	1	25	N/A
600 Grit Ground Glass Diffuser	Thorlabs Part#: DG05-600	1	13	N/A
645 nm Long-pass Colored Glass Filter	Thorlabs Part#: FGL645	2	28	N/A
360-580 nm Colored Glass Bandpass Filter	Thorlabs Part#: FGB39	1	28	N/A
Total:			**188**	**2.3**

For the evaluation of the sensor described in the following sections, the distance between cameras, the baseline T_x, was set to be 80 mm. Unless otherwise noted, a resolution of 1280×720 was used in order to evaluate the full capability of the sensor.

2.10 Flat Plane Experiment

In order to evaluate the output of the sensor, a flat plane was measured. The measurements were taken at a distance of about 0.6 m. Three tests were completed, and the measurements were compared to a best fit flat plane. The root mean squared error ($RMSE$) is calculated as

$$RMSE = \sqrt{\frac{\sum_{n=1}^{N}(e_n)^2}{N}} \tag{2.10}$$

where e_n is the distance from each point to the best fit plane, and there are N points in the point cloud.

First, in order to show the improvement that the speckle system provides to the system, no speckle pattern was used on the flat plane, and Figure 2.12 shows the measurements collected in this experiment, including the raw images in Figure 2.12 (a)-(b), the disparity map in Figure 2.12 (c), the 3D reconstruction in Figure 2.12 (d), and the error map compared to the best fit plane in Figure 2.12 (e). Notably, the long-pass filters were removed from the camera for this experiment. The disparity map seen in Figure 2.12 is essentially meaningless and demonstrates the inability of stereo vision block matching algorithm to find true correspondences between images, ultimately resulting in an RMSE of 141 mm between the 3D reconstruction and best fit plane.

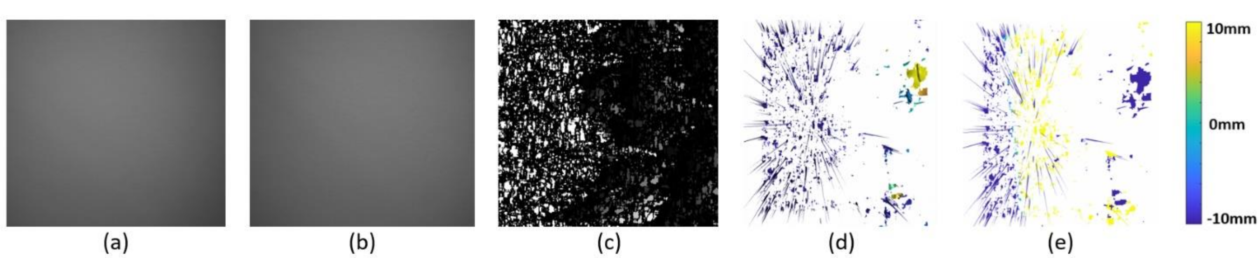


Figure 2.12: Measurement of a flat plane with no speckle pattern projection. (a) Left image; (b) Right image; (c) Disparity map; (d) 3D reconstruction; (e) Error map compared to best fit plane (RMSE = 141 mm).

Next, a measurement of the same flat plane was captured with a single speckle projection system in use, and the data from this experiment is collected in Figure 2.13, including the raw images in Figure 2.13 (a)-(b), disparity map in Figure 2.13 (c), 3D reconstructed data in Figure 2.13 (d), and an error map compared to the best fit plane in Figure 2.13 (e). The RMSE for this test was found to be 1.47 mm, demonstrating that the proposed system can measure the plane with reasonable accuracy.

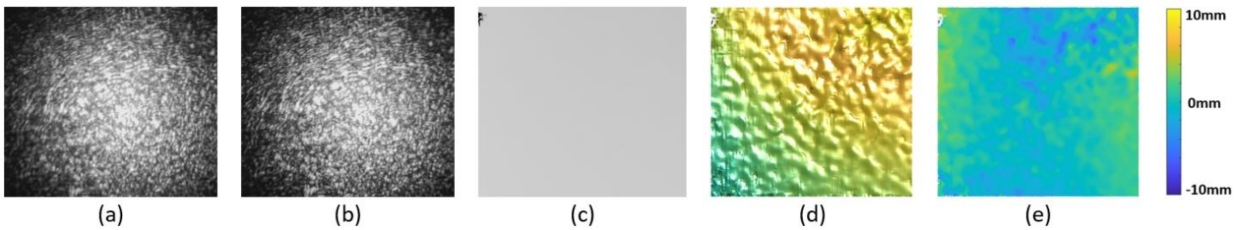


Figure 2.13: Measurement of a flat plane on which a single speckle pattern is projected. (a) Left image; (b) Right image; (c) Disparity map; (d) 3D reconstructed mesh; (e) Error map compared to best fit plane (RMSE = 1.5mm).

Third and finally, to ensure multiple projection systems could be used on the same space, two speckle patterns were projected from two different systems onto the flat plane. The setup for this experiment with two laser speckle projection systems and the cameras is shown in Figure 2.14.

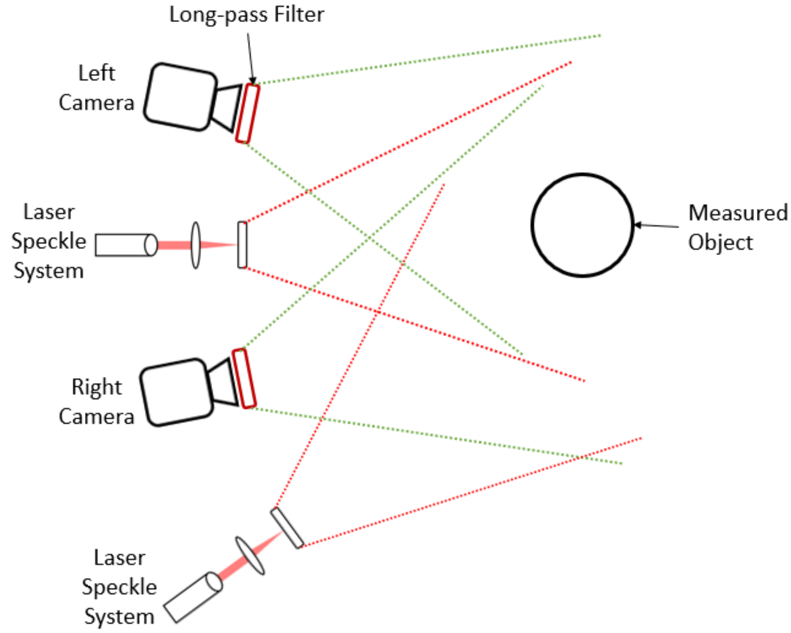


Figure 2.14: Experimental setup for the stereo vision system with two laser speckle projection systems.

The data taken from this experiment is displayed in Figure 2.15, including the raw images in Figure 2.15 (a)-(b), the disparity map in Figure 2.15 (c), the 3D reconstructed data in Figure 2.15 (d), and the error map compared to the best fit plane in Figure 2.15 (e). For this measurement, an RMSE of 1.80 mm is measured, which provides confidence that with two speckle pattern projections the system can still measure the plane with reasonable accuracy.

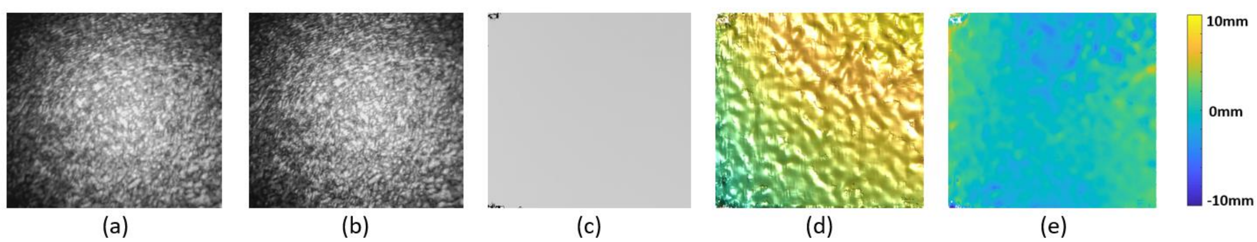


Figure 2.15: Measurement of a flat plane on which two speckle patterns are projected. (a) Left image; (b) Right image; (c) Disparity map; (d) 3D reconstructed mesh; (e) Error map compared to best fit plane (RMSE = 1.8mm).

From these experiments, a few observations can be derived. The laser speckle pattern is proven to be necessary to find stereo correspondences for objects without significant color and texture variation, as the measurement with no speckle pattern projection appears to have no accurate 3D reconstructed data. The system does properly measure a flat plane without significant distortions or major errors, as the RMSE is under 2 mm for both tests with a speckle pattern. There is still some error in the measurements using the speckle pattern, which can be seen on the error maps. However, there is no significant change in result based upon adding an extra speckle pattern to the scene, as the increase in RMSE is just 0.3 mm. The error between setups will be quantified more robustly in Section 2.11.

2.11 Sphere Measurement

Next, the accuracy of the sensor was evaluated by measuring a sphere with a radius of 15 cm. In this experiment, the effect of applying the subpixel matching algorithm was tested, as well as the effect of projecting a second speckle pattern onto the sphere. Each measurement was reconstructed to 3D, and the best fit sphere was computed. Figure 2.16 shows one of the measurements taken with a single speckle pattern projection with a camera resolution of 1280×720, including the left and right images in Figure 2.16 (a)-(b), the disparity map in Figure 2.16 (c), and the 3D reconstruction from integer disparity and sub-pixel disparity in Figure 2.16 (d)-(e).

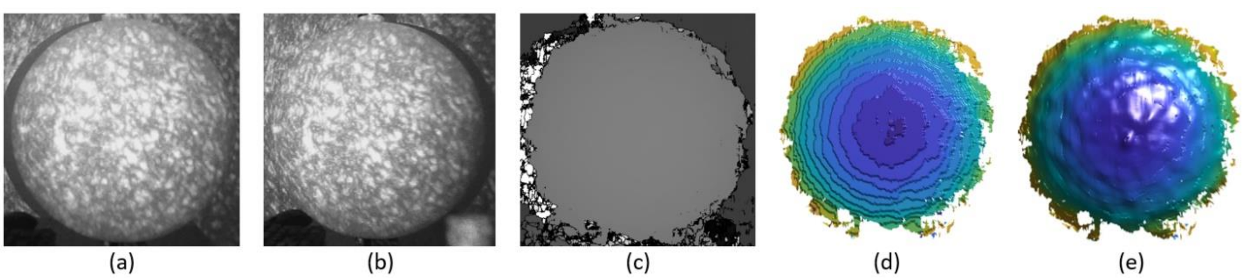
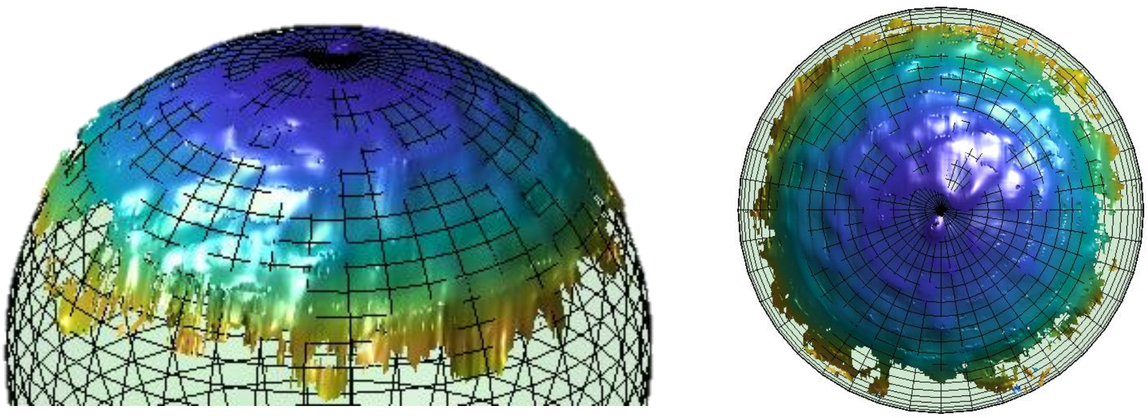


Figure 2.16: Measurement of a sphere at a camera resolution of 1280×720 on which a single speckle pattern is projected. (a) Left image; (b) Right image; (c) Disparity map; (d) 3D reconstruction from integer disparity map; (e) 3D reconstruction from sub-pixel disparity map.

To compute the best fit sphere, a function is used that guesses the center of the sphere, and then finds the error in radius of each point compared to the guessed center point. By summing the errors in radius of all the points compared to the ideal sphere with a radius of 15 cm, a total error can be computed. Many iterations can then be computed to adjust the predicted sphere center until the error is as small as possible. This process proved to effectively locate the best fit sphere, and Figure 2.17 shows the best fit result for the measurement taken in Figure 2.16.



(a) (b)

Figure 2.17: Best fit sphere for 3D reconstructed data. (a) Side view; (b) Straight-on view.

For this measurement, Figure 2.18 (a)-(b) shows error maps of the 3D reconstructed spheres compared to the best fit, ideal sphere.

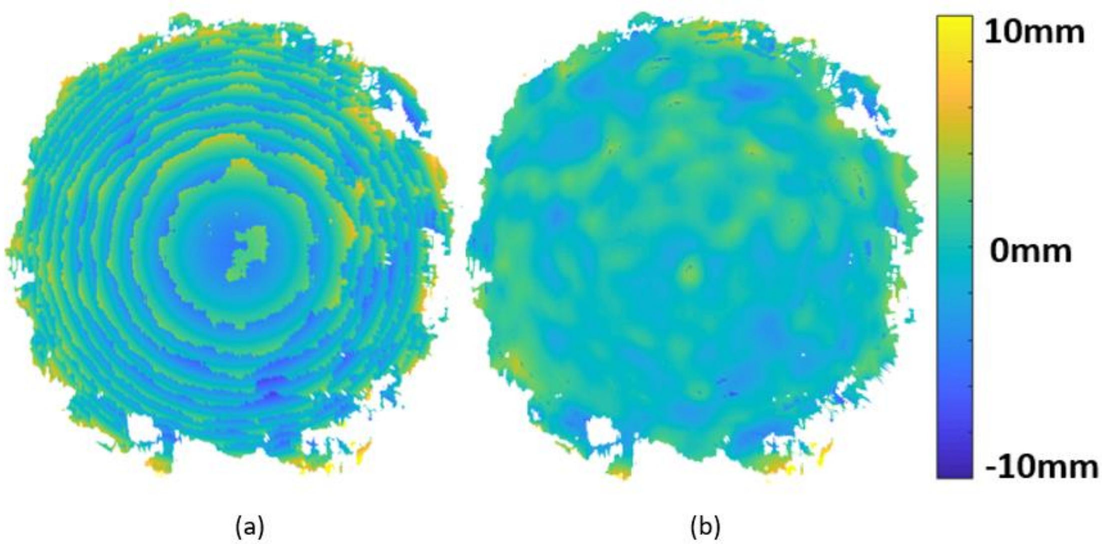
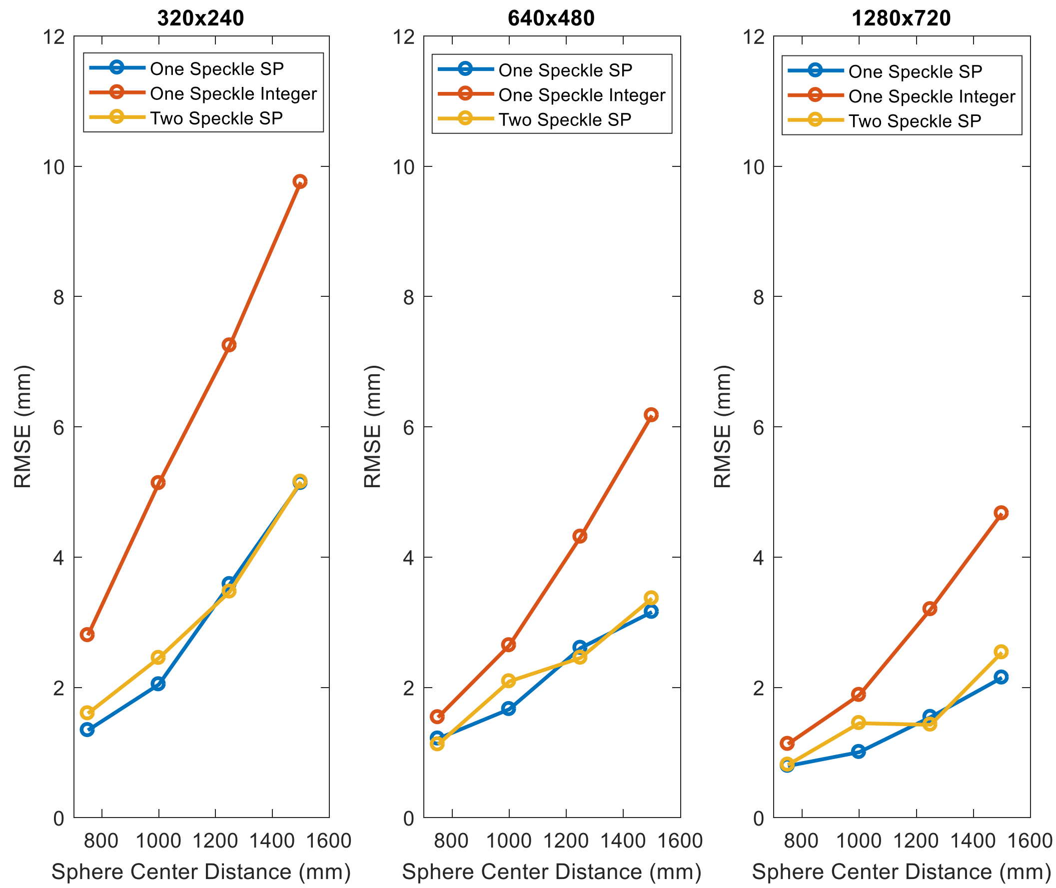


Figure 2.18: Error maps compared to best fit sphere from sphere 3D reconstructed data. (a) Integer disparity (b) Subpixel disparity.

In order to characterize the error, spheres were tested at distances ranging from 0.75 m to 1.5 m. For each measurement, the root mean squared error

$$RMSE = \sqrt{\frac{\sum_{n=1}^{N}(r_n - r)^2}{N}} \tag{2.11}$$

was calculated for the measured area of the sphere, where a point in the point cloud is a distance r_n away from the center of the best fit sphere of radius r, and N such points are used in the measurement. Tests were completed at the camera resolutions of 320×240, 640×480, and 1280×720, and for each test a single speckle pattern was used for one measurement and a double speckle pattern was used for a second measurement. Furthermore, for the single speckle pattern the integer disparity was also tested without using the sub-pixel matching algorithm to evaluate the value added from implementing the algorithm. The results for this experiment are collected in Figure 2.19.



(a) (b) (c)

Figure 2.19: Root mean squared error results for tests at distances between 0.75m and 1.5m and using a single speckle pattern as well as two speckle pattern projections. The single speckle projection is evaluated using both a 3D reconstructed integer disparity map and a 3D reconstructed sub-pixel disparity map. (a) 320×240 resolution; (b) 640×480 resolution; (c) 1280×720 resolution.

From this experiment, there are several important observations. First, it can be seen that the sub-pixel matching algorithm significantly improves the measurement in comparison with the block matching algorithm, which yields integer disparity values. For the twelve tests in Figure 2.19, the RMSE is an average of 1.9 times worse if the sub-pixel matching is not applied. Additionally, there is only a minor effect on the error from adding a second speckle pattern projection on the sphere. Averaging all twelve tests, the increase in RMSE from a single speckle pattern to a double speckle pattern is 0.14 mm. The maximum RMSE increase among all the tests was found to be 0.44 mm. Finally, the resolution improvement from 320×240 to 640×480 in Figure

2.19 (a)-(b) yields a decrease in error by a factor of 1.3. The increase in resolution from 640×480 to 1280×720 in Figure 2.19 (b)-(c) improves the accuracy by a factor of 1.6.

2.12 Complex Object Experiment

Next, a statue with complex surface geometry was measured to evaluate the sensor's ability to capture data on a surface with smaller features. The measurement is shown in Figure 2.20, which shows the images taken in Figure 2.20 (a)-(b), an unfiltered image of the statue in Figure 2.20 (c), the 3D reconstruction from integer disparity and sub-pixel disparity in Figure 2.20 (d)-(e), and the ground truth in Figure 2.20 (f). The ground truth reconstruction was found using a digital fringe projection system with error proven to be much less than the proposed stereo vision system. The system used to take the ground truth data is described in detail by Hyun and Zhang [31].

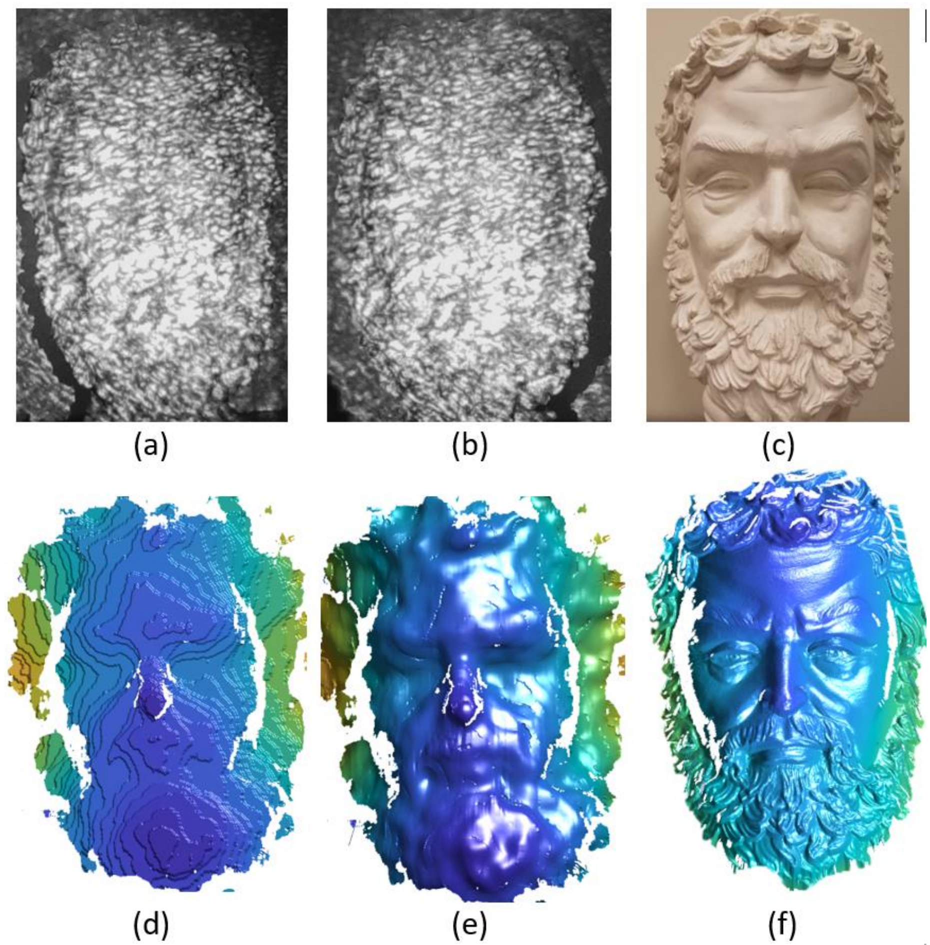


Figure 2.20: Measurement of a statue with a single speckle pattern projection. (a) Left image; (b) Right image; (c) Raw image of statue; (d) Integer disparity 3D reconstruction; (e) Sub-pixel disparity 3D reconstruction; (f) Ground truth.

Importantly, due to the laser speckle projection system, the majority of the face is able to be matched, as can be seen from the disparity map. Only the very steep gradients in the nose area and the sides of the face are not able to be matched. From this data, it is seen that the sensor is able to capture the general shape of the face using the basic block matching algorithm, but facial features are not easily distinguishable. By applying the sub-pixel matching algorithm, facial features such as beard and hair ruffles, geometry around the eyes and eyebrows, and some of the features around the mouth are able to be visually discerned. However, small and sharp details in the face are not able to be captured perfectly, as the face has a smooth appearance.

2.13 Texture Mapping Evaluation

Next, the texture was mapped to a scene to test the texture mapping, as described in Section 2.8. Figure 2.21 shows the texture mapping to the 3D geometry, including an unfiltered color image of the scene and the texture mapped to the 3D geometry.

(a) (b) (c)

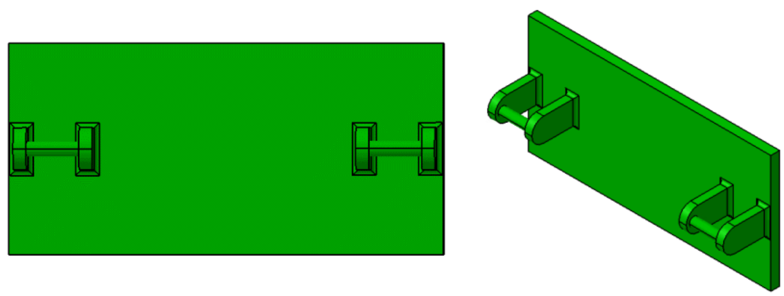
Figure 2.21: Texture mapped to 3D geometry for a scene. (a) Original color of the scene from unfiltered camera; (b) Texture mapped to 3D, direct view; (c) Texture mapped to 3D, offset view.

As a result of this test, the texture content has been shown to be able to be mapped to the 3D geometry.

3. PRECISE AUTONOMOUS VEHICLE NAVIGATION

3.1 Introduction

Based upon the previous chapter, the laser speckle stereo vision system has been shown to be low-cost, low-power, mobile, and accurate. With these characteristics, the system should be useful as a sensor on an autonomous vehicle, particularly for solving the problem of autonomous hitching, which is a crucial process especially for the agricultural and transportation industries. In order to model the hitching problem, an object representing a hitch was designed with the goal of precisely controlling the vehicle to a desired orientation facing the hitch. This target object is shown in Figure 3.1, and has similar features to a standard agricultural three-point hitch. The target was 3D printed in order to rapidly be able to test the design.



(a) (b)

Figure 3.1: Target design for hitch modelling. (a) Perpendicular view; (b) Isometric view.

Importantly, it can be seen that this object has few visual features and little texture. Consequently, the ability to detect this object using classic stereo vision would be very challenging due to the stereo correspondence problem, especially in a real-time, mobile application where computing power is limited. In the following sections, the method for detecting the object and obtaining data will be described in detail. As a preliminary overview, Figure 3.2 shows an overview of major processes needed to control the vehicle, which will be described in more detail in the upcoming sections of this chapter.

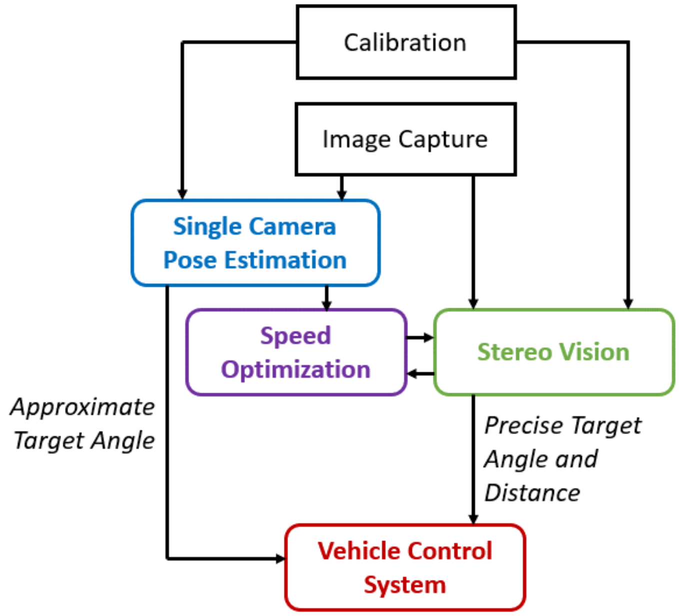


Figure 3.2: Flow diagram of the major processes required for target sensing and vehicle control.

3.2 Hardware Design for System Modelling

The following section details the hardware used specifically for this project in order to model the hitching problem accurately. The model vehicle for this project is the Traxxas Slash 2WD vehicle. This vehicle is large enough to easily fit the cameras, mobile computer, and various batteries and other components on the vehicle. One critical feature of this vehicle is that each wheel has its own suspension, making it a good model for an agricultural vehicle that is designed to be able to navigate rough terrain. The vehicle is also able to drive in reverse, which is very important since most hitching applications require reverse driving. Overall, the vehicle with all the hardware in place is shown in Figure 3.3, which specifically shows the location of the sensor on the vehicle and also the suspension system of the vehicle.

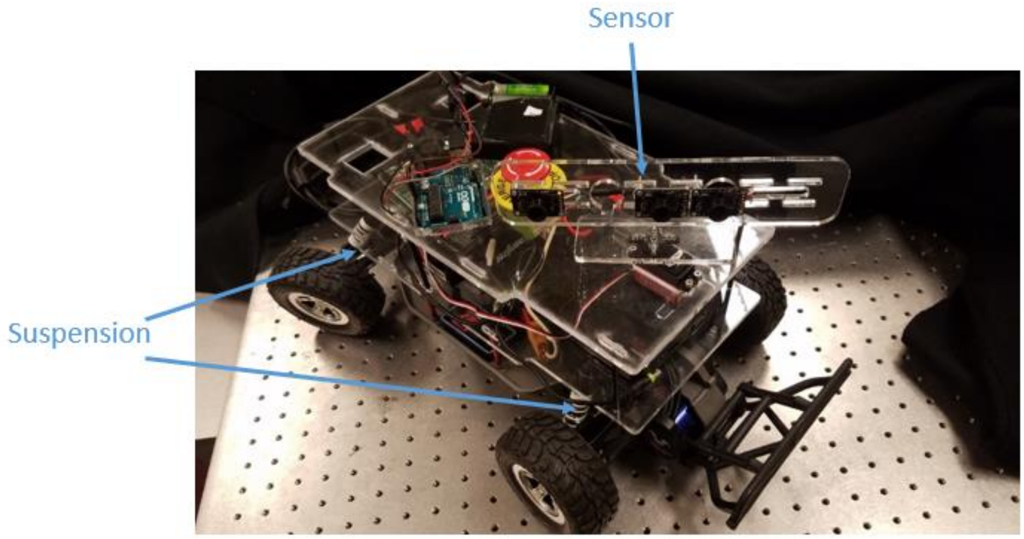


Figure 3.3: Vehicle used to model the problem, showing the location the sensor was placed on the vehicle and the suspension system on the vehicle.

Furthermore, the vehicle has been upgraded with a brushless motor for increased power and ability to control the vehicle speed more precisely. The vehicle is designed by the manufacturer to be controlled by a remote controller, but since the goal is to control it automatically from the onboard computer, an electronic speed controller (ESC) is connected to the motors in order to set the vehicle motion and wheel angle automatically. ESCs are widely used in projects such as DIY electric skateboarding, and are useful for interfacing with the motors on the Traxxas car. The specific controller selected is the VESC, which has open source software called the BLDC tool that creates a GUI running on Qt, which can be used to connect to and control the vehicle motors. The BLDC tool software provides the basic framework for the software used in this project, with new functions and algorithms being added to it as needed.

A mobile computer is used to run the programs developed for this project. The Udoo x86 Ultra was selected as a good mobile computer at a reasonable price and with sufficient computational power. This computer has 4 cores and a 2.56 GHz CPU, in addition to enough USB ports for the cameras used in the project. Furthermore, the Udoo has an onboard Arduino, which is used for controlling a servo motor on the vehicle. The cameras sit attached to the servo motor, so the servo motor can be adjusted to ensure that the object being tracked does not leave the field of view of the cameras. Now that the hardware has been set, a mathematical model for the hardware, particularly the vehicle, can be developed.

3.3 Vehicle Dynamics

Now, since the hardware for modeling the project has been selected, it is important to define the dynamics of the vehicle, in order to be able to ultimately control the vehicle towards the target. First, a coordinate system is defined, which in this application has an origin at the center of the target, with the *x* axis perpendicular to the target. Using this coordinate system, the vehicle orientation can be described. Figure 3.4 shows the coordinate system used to define the vehicle orientation, including the definition of the wheel angle and vehicle angle.

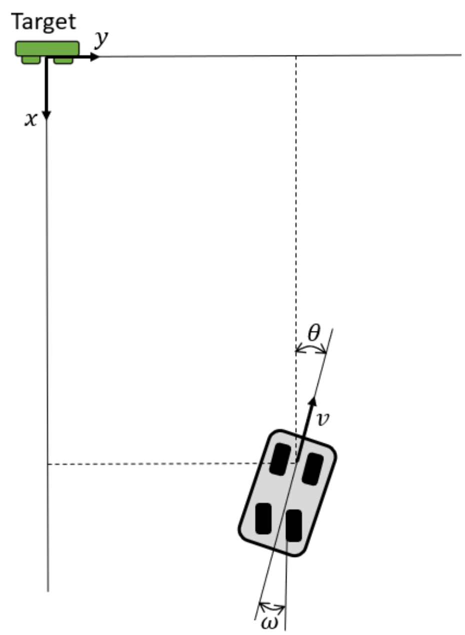


Figure 3.4: Coordinate system used to describe the vehicle orientation, including important vehicle parameters.

For a proper modeling of the system, the rear wheels are the steering wheels, as backwards driving it most common for hitching applications. It is common to simplify the vehicle model by assuming a single wheel in the center of the steering wheels at the same angle of the steering wheels. This simplification is valid since both steering wheels are dependent upon each other and turn at the same angle. This model essentially simplifies the vehicle dynamics to those of a bicycle. Consequently, the vehicle kinematics can be described by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \dot{\boldsymbol{p}} = \boldsymbol{J}\boldsymbol{q} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \tag{3.1}$$

where the angle of the wheel is the parameter ω, and the speed of the vehicle is denoted by v [32]. Since the vehicle has been modeled mathematically, the model can be used in developing a controller. First, however, system identification must be done to relate the actual vehicle physical dynamics with the corresponding inputs to the controller.

3.4 System Identification

The two vehicle parameters that are able to be controlled are the velocity v of the vehicle in cm/s and the angle of the wheels in radians, ω. In order to control the system, it must be known what signal must be passed to the vehicle through code in order to correspond to a meaningful change in a meaningful unit on the vehicle. The signal passed to change the wheel angle, ω_{duty}, is a signal between zero and one. A variety of duty cycles between zero and one were sent to the wheels, and the wheel angle was measured for each signal that was sent. Upon investigation, relationship between ω_{duty} and ω was found to be roughly linear, and the best fit line for this relationship was found to be

$$\omega_{duty} = 1.05 * \omega + 0.39. \tag{3.2}$$

Next, the relationship between the physical velocity v and the input value v_{duty} was able to be determined. For several inputs, the velocity v was measured by timing how long it took the vehicle to move a set distance. The relationship between v_{duty} and v was also found to be linear. After fitting a line to the data points, the relationship was found to be

$$v_{duty} = 0.000692 * v + 0.00934. \tag{3.3}$$

As a result of the system identification, the physical velocity and physical wheel angle are able to be set by the controller by calculating the duty cycle input required to set the vehicle to the desired velocity or wheel angle.

3.5 Near Distance Control System

With the vehicle dynamics defined, a control system can be developed for the system. A stable control system proposed by Kanayama et al. is implemented in order to control the vehicle position and angle [33].

In the application, it is desired for the vehicle to approach the target perpendicularly.

Consequently, the reference, or desired values, for y and θ are both zero, so the measured value of these parameters is also the error. The control scheme can then be described by

$$\omega(K_y, K_\theta) = \omega_r + v_r\big(K_y y + K_\theta \sin(\theta)\big), \tag{3.4}$$

where K_y and K_θ are constant control parameters and ω_r is the feedforward wheel angle, which is set to zero since the desired path is a straight line perpendicular to the target [33]. Note that as a simplification, the velocity v of the vehicle is set as a constant, so control of the velocity is not necessary. Overall, the system is modeled with the block diagram given in Figure 3.5, which shows the feedback control system.

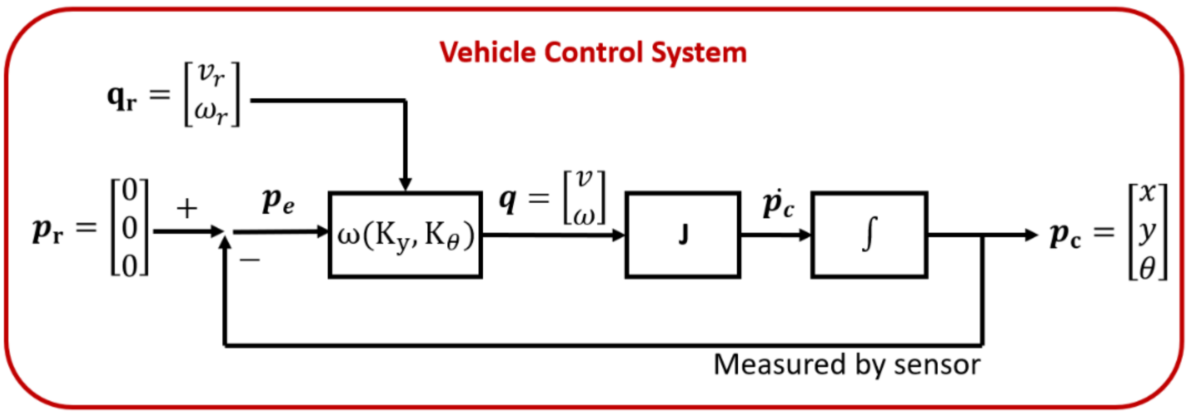


Figure 3.5: Block diagram for the vehicle feedback control system.

In practice, the control constants can be chosen strategically, and some research has been done on what constants have good results [33]. The damping ratio for chosen constants is

$$\zeta = \frac{K_\theta}{2\sqrt{K_y}}. \tag{3.5}$$

3.6 Target Identification with Single Camera

With the vehicle dynamics defined and the control system developed, it is needed to obtain the signals from the sensor that can identify the position and orientation of the target relative to the vehicle. As described in Section 2.8, a texture camera has been added to the vision system, and it can be leveraged to perform several helpful image processing algorithms. This camera has been calibrated in order to determine its intrinsic and extrinsic parameters.

An important step throughout the control process for the vehicle is identifying and tracking the target. There are many methods for object identification and tracking, but for this application the method of color thresholding has proven to be sufficient. The color of the object is characterized by its RGB values in an image. For each pixel in the image, it is determined if the color of that pixel is within the range of RGB values corresponding to the target color. A binary threshold image of white and black pixels is created, with white corresponding to where the color matches the object color. Following this, erosion and dilation algorithms are applied [18]. The erosion algorithm scans the binary image and reduces the size of the white clusters of pixels. Importantly, this means that small, noisy clusters will be removed entirely from the image. The dilation algorithm is then applied, which scans the object and reduces the size of the black clusters of pixels, essentially the opposite of the erosion algorithm, ensuring that any small areas of the target that are not found to be within the threshold color values still are filled in. This entire process can be visualized in Figure 3.6, which also shows how the algorithm is robust to other objects of similar shades of color to the target.

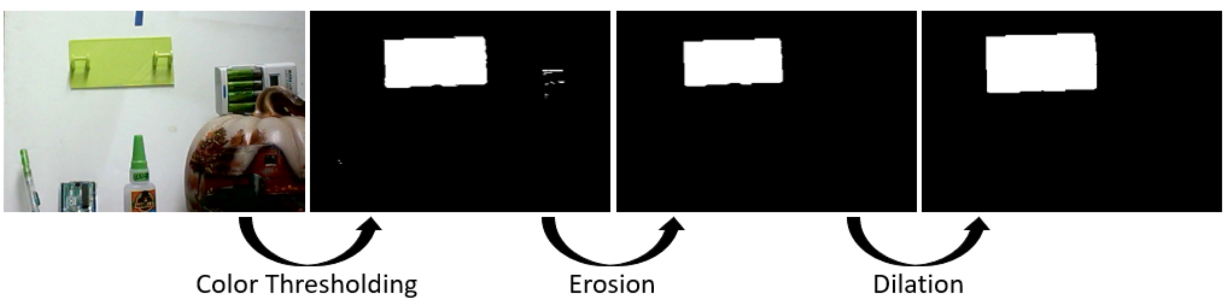


Figure 3.6: Example scene demonstrating algorithms used for target identification.

For the target under investigation, it would be helpful to know its corner locations in the image. If the corner locations are known, the pose of the object will also be known, and several optimizations of the speed of the system will be able to be incorporated. While simple corner finding algorithms exist, a more robust two-step approach has been identified and utilized. First, the findContours algorithm is used to find the edge of the target in the binary image, which is based upon the algorithm and methods proposed by Suzuki [18] [34]. Then this contour is approximated as a quadrilateral using the Douglas-Peucker algorithm, which is one of the most widely used algorithms for reducing the number of vertices to represent a contour, and OpenCV implements this algorithm with approxPolyDP [18] [35]. Consequently, the corners of the quadrilateral correspond to the approximate corners of the target in the image. For some uses, these corners are sufficient as an approximation of the corner locations of the target. However, in other cases more precise corner locations may be desired. In this case, the corners can be refined using the cornerSubPix algorithm [18]. Using the known approximate corner locations in the binary image, the cornerSubPix algorithm can be applied on the original image in grayscale in order to more precisely identify the corners of the target. The entire process of corner identification, from finding the contours, to quadrilateral fitting, to sub-pixel corner refinement, can be visualized in Figure 3.7.

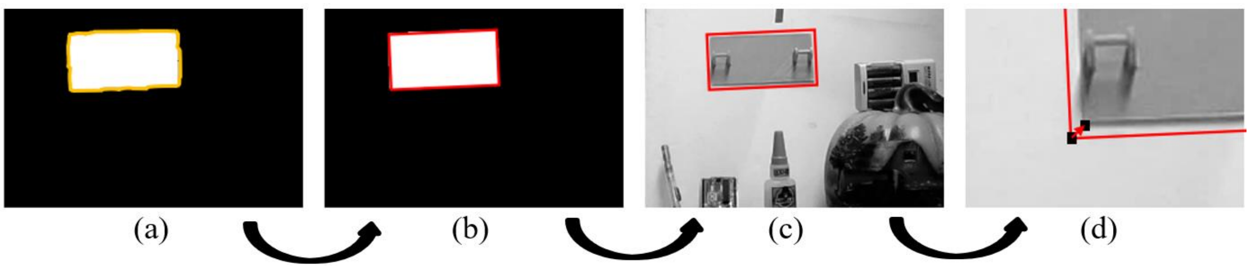


Figure 3.7: Flow of the process of finding the corner locations of the target. (a) Contour finding on the binary image; (b) Conversion of the contour to a quadrilateral; (c) Quadrilateral visualized on the grayscale image; (d) Effect of using sub-pixel corner refinement algorithm on the grayscale image to precisely find the corner of the target.

3.7 Target Position and Angular Estimation using Single Camera

By using the corner locations along with the camera intrinsic calibration, it is possible to identify the position and orientation of the target. While this method is less accurate than stereo vision at short distances, it is accurate enough for control of the vehicle at far distances where the stereo vision is less reliable, particularly for angular estimation. Furthermore, knowing the rough distance allows for several speed improvements to the algorithms.

The problem of using known 2D pixel coordinates from an image along with known 3D geometry of the corresponding object in order to calculate the position and orientation of the object has been solved with various methods. In this project, the EPnP method has been incorporated using the solvePnP OpenCV algorithm [18] [36]. Using this algorithm, the approximate translation and rotation of the target is found, and since the camera has been calibrated, the data can be translated to the world coordinate system, in accordance with Equation (2.3). The entire process of using the single camera for target identification and position estimation of the target is summarized in Figure 3.8.

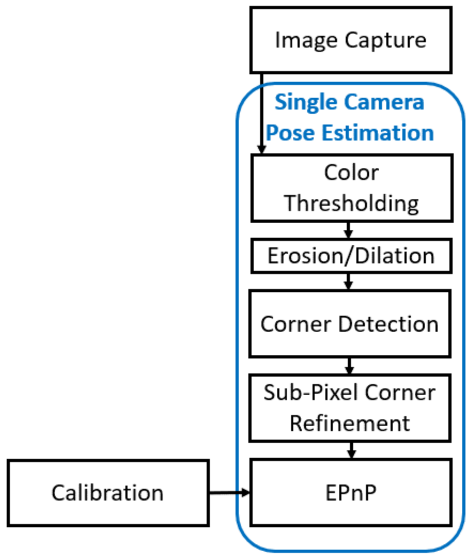


Figure 3.8: Flow diagram of single camera processes and algorithms required for target pose estimation.

3.8 Target Position Estimation using Stereo Vision

With the 2D camera able to identify the object and track its location, it is now possible for the block matching algorithm to be incorporated in order to find the position and orientation of the target more precisely using the stereo vision cameras. The first step in this process is to transform the coordinates of the corner pixels from the single camera to the rectified left image of the stereo camera. By computing this transformation, it will be known which disparity map pixels need to be reconstructed to obtain the position of the target in the 3D world.

To transform the camera coordinates from the single camera to the left camera rectified image, the pinhole camera model should be utilized. For each camera, Equation (2.3) can be used. This yields 6 equations. However, although all of the intrinsic and extrinsic parameters are known, there are still 7 unknowns: the three world coordinates, the two camera coordinates on the left image, and the scaling factor for both cameras. In order to solve one unknown parameter, the estimated position of the target from the single camera EPnP algorithm can be used. Using this information, Equation (2.3) for the single camera can be solved for the scaling factor, s. Then, the entire system of equations for both cameras has 6 equations and 6 unknowns, so it is solvable. Consequently, the pixel coordinates of the corners of the target are known in the left image of the stereo camera. By repeating the process used in the above paragraph, the corners of the target can be transformed from the left image to the rectified left image. Notably, the intrinsic and extrinsic matrices for the rectified camera are outputs of the stereoRectify function of OpenCV [18]. Finally, the coordinates of the corners of the target on the left rectified image are known. Using these coordinates, the 3D reconstruction can be applied for only the pixels in the integer disparity map that are on the target. Consequently, the target 3D position can be known, and can be relayed to the vehicle controller.

3.9 Target Angle Estimation using Stereo Vision

In controlling the vehicle to a precise position and orientation, it is important for the stereo camera to not only calculate the position of the target, but also its orientation. However, the raw 3D reconstruction of integer disparity values is insufficient for any angular calculation, due to the problem of depth resolution discussed in Section 2.6. Using the sub-pixel processing algorithm proposed in Section 2.6, certain portions of the target can be processed with high accuracy and the angle of the target can be calculated. In looking at the target, it can be recognized that the angle of the target can be found by finding the angle of a line of pixels within the flat area of the target. For example, Figure 3.9 shows several lines of pixels. After reconstructing one of these lines to a 3D point cloud, the best fit line can be calculated. The analysis can be simplified by only considering one plane parallel to the floor in the 3D world and therefore neglecting the vertical Y^w component. This is possible since only the angular rotation with respect to this plane is relevant to the vehicle control. The slope of the best fit line with n points in the X^w-Z^wplane is defined as

$$m = \frac{n * (\sum_{i=1}^{n} x_i * z_i) - (\sum_{i=1}^{n} x_i)(\sum_{i=1}^{n} z_i)}{n * (\sum_{i=1}^{n} x_i^2) - (\sum_{i=1}^{n} x_i)^2}. \tag{3.6}$$

The arctangent of the slope is the angle α of the line in radians with respect to the Z axis of the left camera, described mathematically as

$$\alpha = \tan^{-1} m. \tag{3.7}$$

By averaging the angle calculated from several lines, such as five lines in Figure 3.9 representing the pixels that would be sub-pixel processed to find the target angle, the noise of the measurement can be reduced to a stable signal that can be used as a controller input. The location of the lines is able to be determined based upon the corner location that previously was completed for object identification. Consequently, the position and orientation of the target are known, so the vehicle can be controlled to a precise position relative to the target.

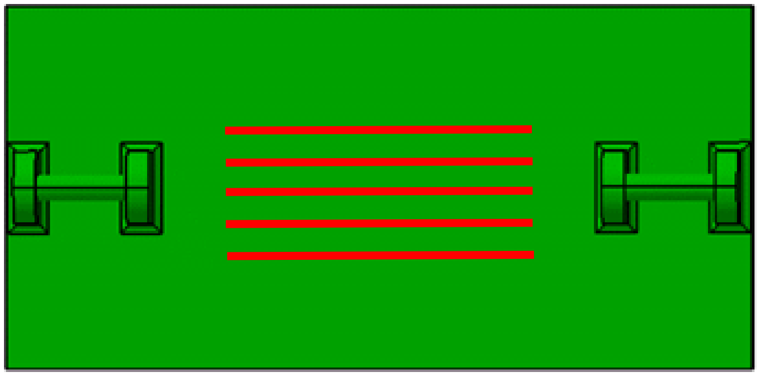


Figure 3.9: Red lines on target representing lines of pixels to be sub-pixel processed for target angle estimation.

3.10 Moving Average Filter

The signals obtained from the cameras can be subject to noise. In particular, the angular estimation from the 2D camera and the 3D camera at larger distances can be noisy. In order to solve this problem, a moving average filter is implemented. The equation for an L point moving average filter is

$$y(n) = \frac{1}{L}\sum_{i=0}^{L-1} x(n-i)\,. \tag{3.8}$$

By incorporating a moving average filter, the noise of the measurements can be reduced, allowing for smoother outputs of the control system described in the next section.

3.11 Speed Improvements

The theory is in place to successfully control the vehicle. However, in practice the limitations of computing power require speed optimizations in order to maximize the effectiveness.

The first way the speed can be influenced is by choosing the resolution of the images. Most significantly, the resolution of the two stereo camera images is very important, as the size not only affects the time for the images to be captured, but also for the block matching to be computed. For vehicle control, sufficient accuracy can be achieved with a 320×240 resolution for the stereo camera, based upon the accuracy results from Section 2.11. Next, the single camera resolution can be chosen. The resolution can be higher for better corner identification of the target, because less processing is required for this image. Therefore, the single camera resolution is chosen to be 640×480.

A multithreading approach can also be used to distribute computer resources, resulting in faster processing. The Udoo x86 Ultra computer used in this project can process four threads simultaneously. Figure 3.10 shows the overall multithreading of the image capturing, processing, and vehicle control process. This demonstrates how there are never more than four threads in use at the same time. Furthermore, there is limited downtime where only a single thread is being used (note that the block matching algorithm is designed to use multiple threads at once, if available).

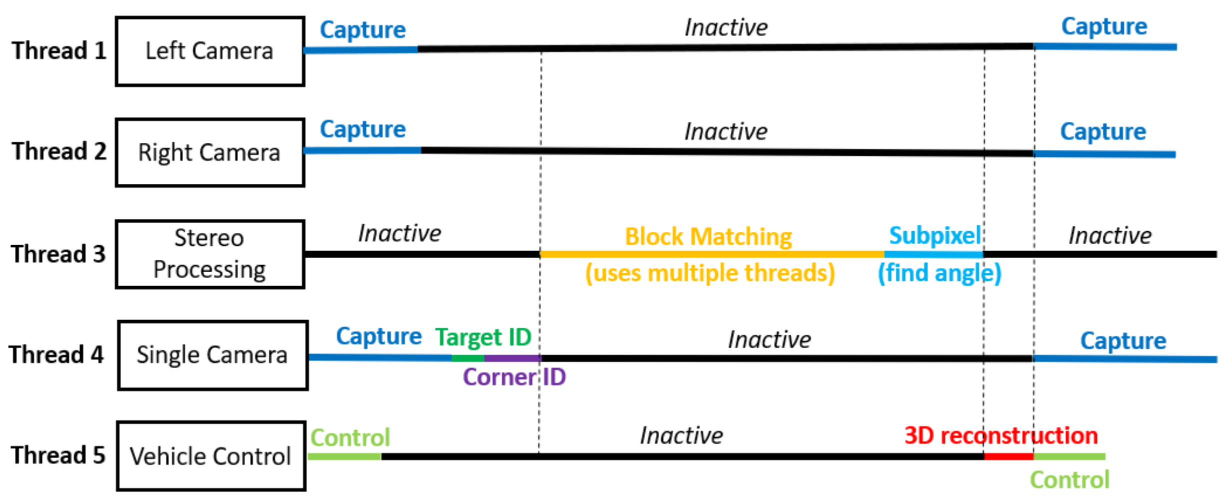


Figure 3.10: Summary of thread usage in multi-threading scheme.

Additionally, the block matching algorithm can be sped up considerably by a couple of constraints. Since the coordinates of the corners of the target in the rectified left image are known based upon Section 3.8, the disparity is able to be processed using the block matching algorithm only for the portion of the image corresponding to the target. As a consequence of this constraint, large portions of the image do not need to be processed at all, substantially reducing computation time. As an extension of this speed improvement, when the vehicle is very close to the target, the left and right edges of the target do not need to be reconstructed, only the middle of the target must be reconstructed, since the target takes up most of the camera field of view. Importantly, this prevents any change in measurement if a small portion of the target leaves the field of view of the left or right camera during the control of the vehicle at a close distance, since the edge of the target is not processed in the disparity map.

Another important constraint is using depth estimation to limit the disparity range that must be searched between images. As explained in Section 2.5, the block matching algorithm searches along the line of pixels in the rectified right image in order to find the best matching window for the window in the rectified left image. By constraining the disparity range, only a small number of windows in the rectified right image need to be searched in order to find the correct match to the rectified left image. This range can be set variably depending on how far away the object is, if an estimation can be obtained. If an object is a distance Z^c away from the stereo camera with baseline T_x along the Z^c axis in the pinhole camera model, the disparity δ corresponding to that object is known to be

$$\delta = \frac{fT_x}{Z^c}. \tag{3.9}$$

since all other variables are known parameters following calibration [18]. Therefore, if a reasonable estimation of Z^c can be computed, the disparity can be constrained, so that less computation time is needed. Notably, in this exercise Z^c is equivalent to Z^w, since the left camera coordinate system is also the world coordinate system.

Two methods for finding an estimation of Z^c exist. Initially, the translation estimation obtained from the single camera EPnP algorithm can be used. This is a reliable enough estimate to find the approximate disparity. Another method is to use historical results from the previous disparity map. This is effective, especially at low to medium vehicle speeds. Using historical data, the disparity range can be reduced to 16, which is the minimum allowed in the stereoBM class [18]. At this disparity range, the data is just as reliable as if a larger range was used, but the computation time is significantly reduced. In summary of the speed improvement methods that constrain the block matching algorithm, Figure 3.11 is provided.

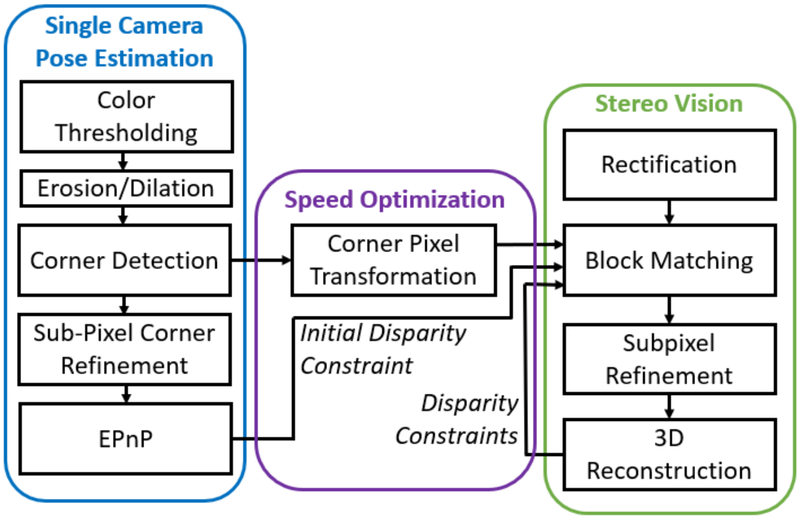


Figure 3.11: Flow diagram showing the speed optimization process in coordination with the image capture and processing algorithms.

3.12 Far Distance Control System

The control system described in Section 3.5 is an effective control system at close range, which will be evidenced in a later section. Additionally, it will be shown to have a high degree of accuracy at a starting angle of approach of up to 15° from the line perpendicular to the target in either direction, from a starting distance of 1.5 m from the target. However, a wider angle of approach may be desired, in order for the system to operate starting from a more arbitrary starting position. A control algorithm has been developed to solve this issue. The idea of this algorithm is that the vehicle will move from the starting position toward the target center line at an angle. The algorithm is practically implemented with the equation

$$\omega = c * \beta + \omega_0 \tag{3.10}$$

where ω is the wheel angle set to control the vehicle, ω_0 is the wheel angle when the vehicle is oriented directly facing the target for which a good value is able to be found with testing, c is a scaling constant for which a good value is able to be found with testing, and β is the angle of the vehicle with respect to the target, as shown in Figure 3.12.

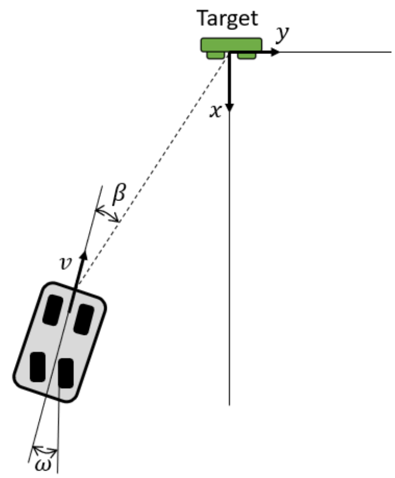


Figure 3.12: Relevant parameters for the control of the vehicle at a far distance.

Importantly, β can be obtained from knowing only the position of the target relative to the camera, meaning it is not dependent upon the actual angle target with respect to the vehicle. Therefore, only the position of the target in the camera and the angle of the servo motor with the cameras on it is needed for this computation. Deciding the value of ω_0 does depend on the angle however. If the vehicle is to the left of the target at an angle of greater than 15º, then ω_0 should be a positive constant, while if the vehicle is to the right of the target at an angle of greater than 15º, ω_0 should be a negative constant. Using this methodology essentially eliminates the effect of noise from the single camera angle estimation, as long as the estimation is accurate enough to determine whether the angle of the target is between ±15º, less than -15º, or greater than 15º.

This control system can successfully maneuver the vehicle from up to 35º away from the line perpendicular to the target to within 15º of the target at a distance of 1.5 m away. Figure 3.13 shows an example of approximately the way the vehicle moves from a larger angle to end within ±15º of perpendicular to the target. At 1.5 m, the control system from Section 3.5 takes over, and the vehicle is precisely maneuvered to the target. If the vehicle is within ±15º and at a distance of larger than the 1.5 m, it just moves straight towards the target.

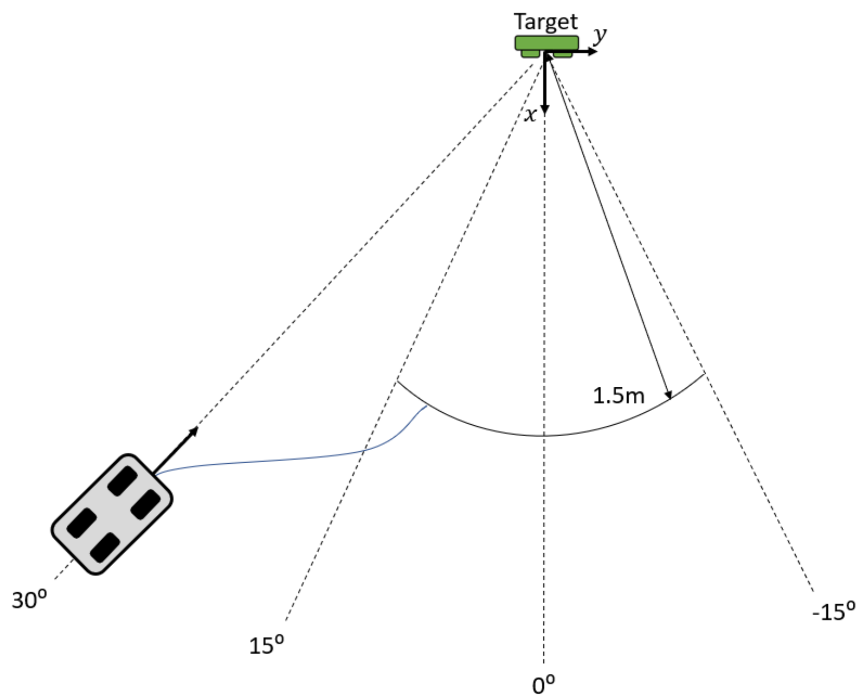


Figure 3.13: Example path of the vehicle into the ±15º range using the far distance control system.

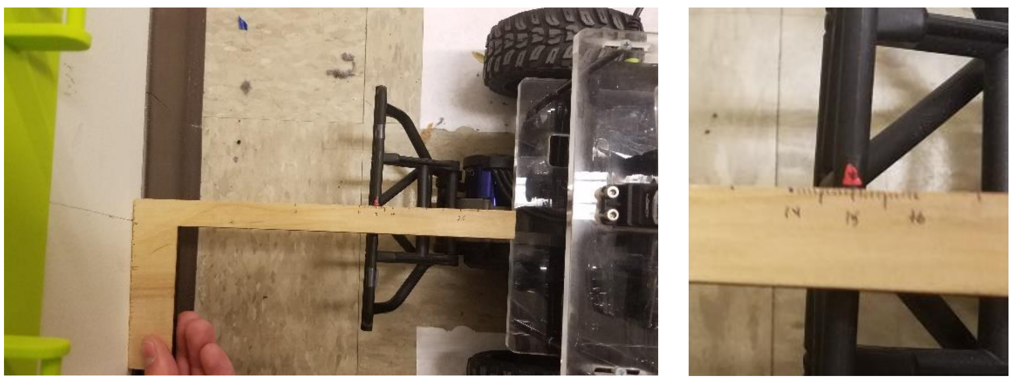
This system has proven to work effectively, provided the vehicle starts at least 2.5 m from the target. This gives sufficient space for the vehicle to move to within 15º of the target.

Based upon this algorithm, the vehicle is able to be successfully controlled to the target as long as it is within a 70º wide cone, a large enough space that a vehicle could presumably be controlled to within this range with relative ease in a real-world application of this system.

3.13 Vehicle Testing

With the vision system developed and incorporated onto the vehicle and a control system adapted to precisely control the vehicle, the entire system can be examined through testing. Since the far distance control system from Section 3.12 is effective at moving the vehicle to within 15º to perpendicular from the target at 1.5 m, the testing is done starting at 1.5 m between -15º and +15º of the perpendicular line of the vehicle. Several iterations of testing were completed to evaluate different control schemes and improvements. In this section, three different schemes will be explained and evaluated experimentally.

The system was tested by starting the vehicle at various angles of approach, and then measuring the final orientation of the vehicle after it stopped. In order to measure the final vehicle position, a ruler, protractor, and squared board, or square, were able to be used to measure the relative vehicle position, using the coordinate system from Figure 3.4. The target was placed on a wall, so the wall could be used as a perpendicular surface to measure from. Figure 3.14, Figure 3.15, and Figure 3.16 show the method used to measure each component. The x and y components are measured to the nearest 1 mm relative to a dot placed on the vehicle front bar using a ruler and the square, while the θ of the vehicle is measured to the nearest 0.5° relative to the front bar on the vehicle using a protractor and square. Each figure also shows a zoomed in view of the measurement.



(a) (b)

Figure 3.14: Measurement of the x coordinate of the vehicle position. The point measured on the vehicle is the black dot on the black bar, and for the measurement shown, a value of 150 mm was measured. (a) The square is set perpendicular to the wall and target; (b) Zoomed-in view of the measurement.

(a) (b)

Figure 3.15: Measurement of the y coordinate of the vehicle position. The point measured on the vehicle is the black dot on the black bar, and for the measurement shown, a value of 14 mm was measured. (a) The square is set perpendicular to the wall and target, and the square is shown aligned with a reference line drawn on the wall; (b) Zoomed-in view of the measurement.

(a) (b)

Figure 3.16: Measurement of the θ of the vehicle position. The reference on the vehicle is the straight black bar, and for the measurement shown, a value of 2.0° was measured. (a) Measurement perpendicular to the wall; (b) Zoomed-in view of the measurement.

Figure 3.17 shows three starting angles of approach that the vehicle was tested at, with the figure showing a starting angle of approach of 15º and example path from the starting position. These angles of approach were tested since the vehicle can be maneuvered to be within 15º of the center line based upon the control system developed in Section 3.12. The tests began at a distance of 1.5 m from the target.

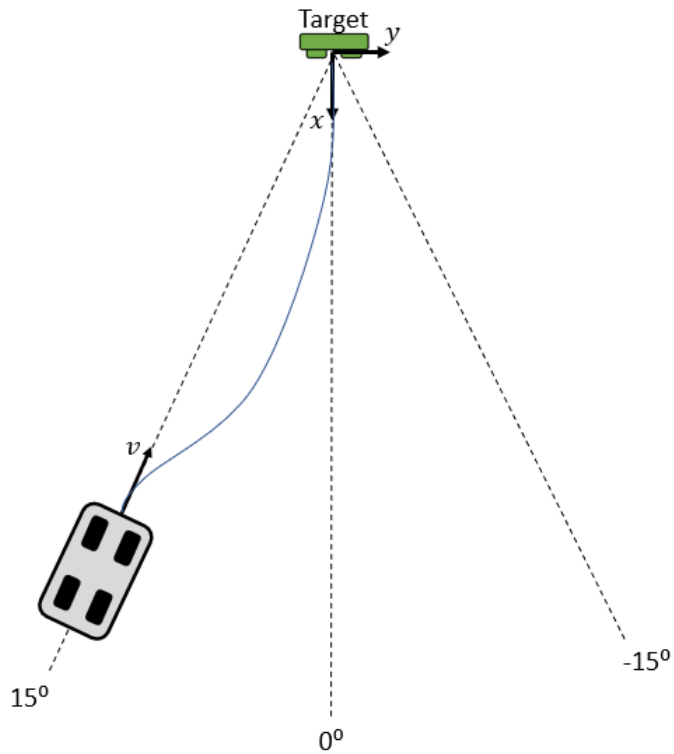


Figure 3.17: Three different angles of approach to the target used for vehicle testing. Shown is the vehicle at a starting angle of approach of 15º.

First, a basic control scheme was implemented with a simple implementation of the controller described in Section 3.5. Two stages of control were used. When the vehicle detected the target at a distance of greater than 1.15 m, as measured by the stereo camera, the vehicle moved straight towards the target. Once the distance was less than 1.15 m, the vehicle began to use the control scheme from Equation (3.4) to control the vehicle. The threshold distance of 1.15 m was chosen since the angular detection with the stereo camera was found to have greater noise at distances greater than about 1.15 m. Once the vehicle reached within 275 mm of the target, the vehicle was stopped. In this scheme, the positional and angular signals were obtained only through the stereo camera. With initial testing, stable values for K_y and K_θ were found to be 0.0010 and 0.060, respectively. According to Equation (3.5), these values correspond to a theoretical damping ratio of 0.95, close to critical damping. The resulting control was stable. The vehicle was tested with the aforementioned method of using only stereo vision, and the results of the tests are provided in Table 3.1, which shows what angle relative to the target the vehicle started at, corresponding to Figure 3.17. The table also gives the individual final orientation error for each test, and then for each angle of approach the final orientation errors are averaged. Notably, the vehicle control is consistent, but not successful in driving the errors of the lateral position and angle of approach to zero. Upon investigation, one main problem that was determined was that 1.15 m was not sufficient distance to complete the vehicle control, and more space would be needed.

Table 3.1: Vehicle testing results using only stereo vision for control. The vehicle starting angle of approach is given for each test, and then the measured final orientation error is given. For each angle of approach, the final orientations are averaged to find the average final orientation error.

Starting Angle of Approach (°)	Final Orientation Error (x (mm), y (mm), θ (°))	Average Final Orientation Error (x (mm), y (mm), θ (°))
-15	(-10, 35, -7) (-10, 45, -8) (-10, 45, -8)	**(-10, 42, -7.7)**
0	(-10, 5, 1) (-5, 2, 1) (5, 3, 1.5)	**(-3, 3, 1.3)**
15	(-5, -35, 9) (-5, -40, 9.5) (-5, -43, 10)	**(-5, -39, 9.5)**

In order to help solve this problem, another stage of control was added. As described in Section 3.7, the angle of the vehicle is able to be known from the single camera, as a result of the implementation of the EPnP algorithm. Furthermore, at larger distances this angle measurement was found to be less noisy than that of the stereo camera. Ultimately, a stage of control was added between 1.15 and 1.5 m, where the single camera measurement was used for the angular signal. As before, when under 1.15 m, stereo vision was used to measure the target angle. Furthermore, regardless of distance, the stereo vision was used to measure the distance from the target. This scheme is shown in Figure 3.18, which summarizes the different areas around the target, and what control scheme and sensing is used in each region, including both the near and far distance control schemes.

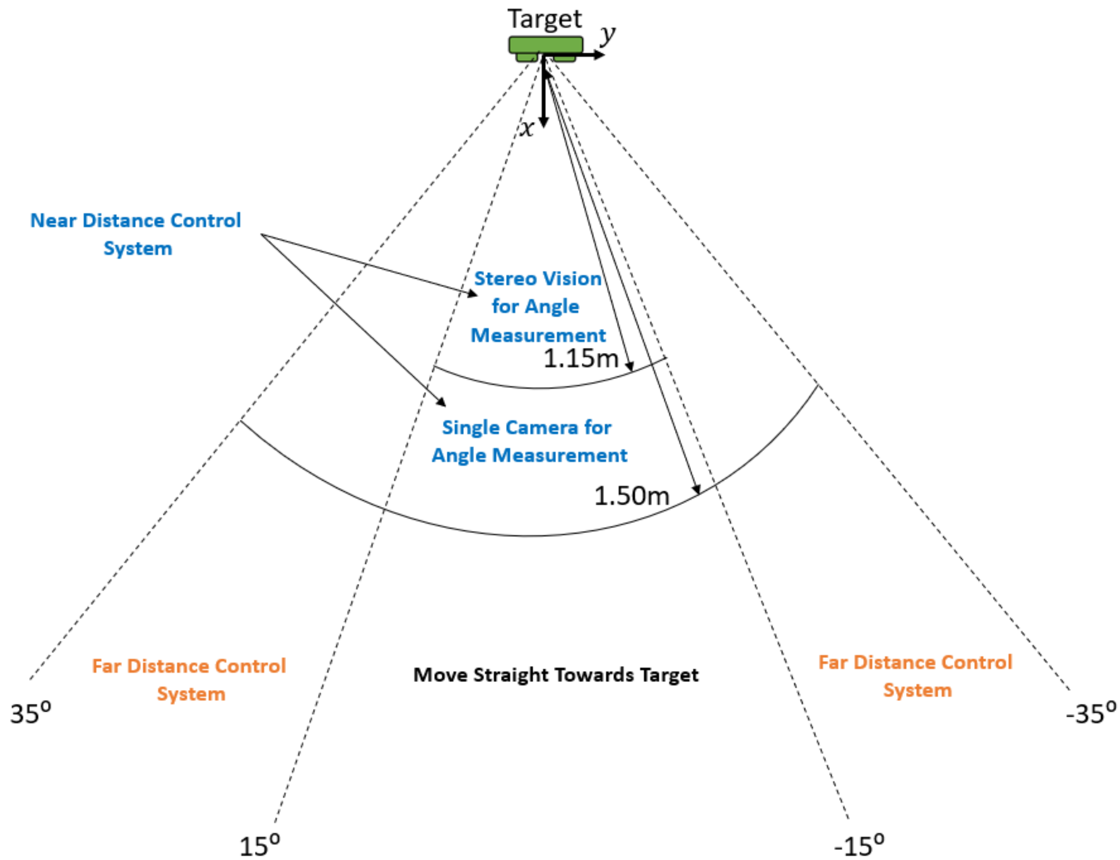


Figure 3.18: Signal used for vehicle control at different vehicle distances from the target. At under 1.15 m, stereo vision is used to measure the target angle, which is the signal for the near distance control system. At 1.15-1.5 m, the single camera is used to measure the target angle, which is the signal for the near distance control system. At greater than 1.5 m between ±15º, the vehicle just moves straight towards the target until it is within 1.5 m. When the vehicle is at a distance of greater than 1.5 m and outside of ±15º, the far distance control system is used to move the vehicle to within ±15º.

With this addition to the previous control scheme, the vehicle was again tested, with significantly better results, but the system was still unsuccessful in minimizing lateral and angular errors. This second test is shown in Table 3.2, including the starting vehicle angle from the target and the measured final orientation error for each test. For each test, the vehicle started at 1.5 m from the target. The average of the tested final orientation errors for each starting angle of approach is also given in Table 3.2.

Table 3.2: Testing results using stereo vision for angular measurement at under 1.15 m and the single camera for angular measurement at between 1.15 m and 1.5 m. The vehicle starting angle of approach is given for each test, and then the measured final orientation error is given for each test. For each angle of approach, the final orientation errors are averaged to find the average final orientation error.

Starting Angle of Approach (°)	Final Orientation Error (x (mm), y (mm), θ (°))	Average Final Orientation Error (x (mm), y (mm), θ (°))
-15	(-5, 25, -3.5) (-5, 20, -3) (-5, 20, -3)	**(-5, 22, -3.2)**
0	(-10, 3, 1) (-5, 0, 1) (0, 0, 2)	**(-5, 1, 1.3)**
15	(0, -15, 6) (-10, -20, 6) (-5, -17, 6)	**(-5, -17, 6.0)**

Upon this testing, it became apparent that modifying the control scheme again could yield significantly better results. From observation, it was recognized that the vehicle could approach the centerline more quickly at the beginning of the control, while at the end of the testing as it approached the target it was much more important for the vehicle to align angularly with the desired angle of approach of the target. Consequently, a three-stage controller was incorporated and tested, with different gains used in each stage. Figure 3.19 (a)-(b) shows the gains found to be effective for this control system, which used all the same input signals as the system described in the previous paragraph, and Figure 3.19 (c) shows the theoretical damping ratio calculated from Equation (3.5). In this control scheme, Figure 3.18 still applies.

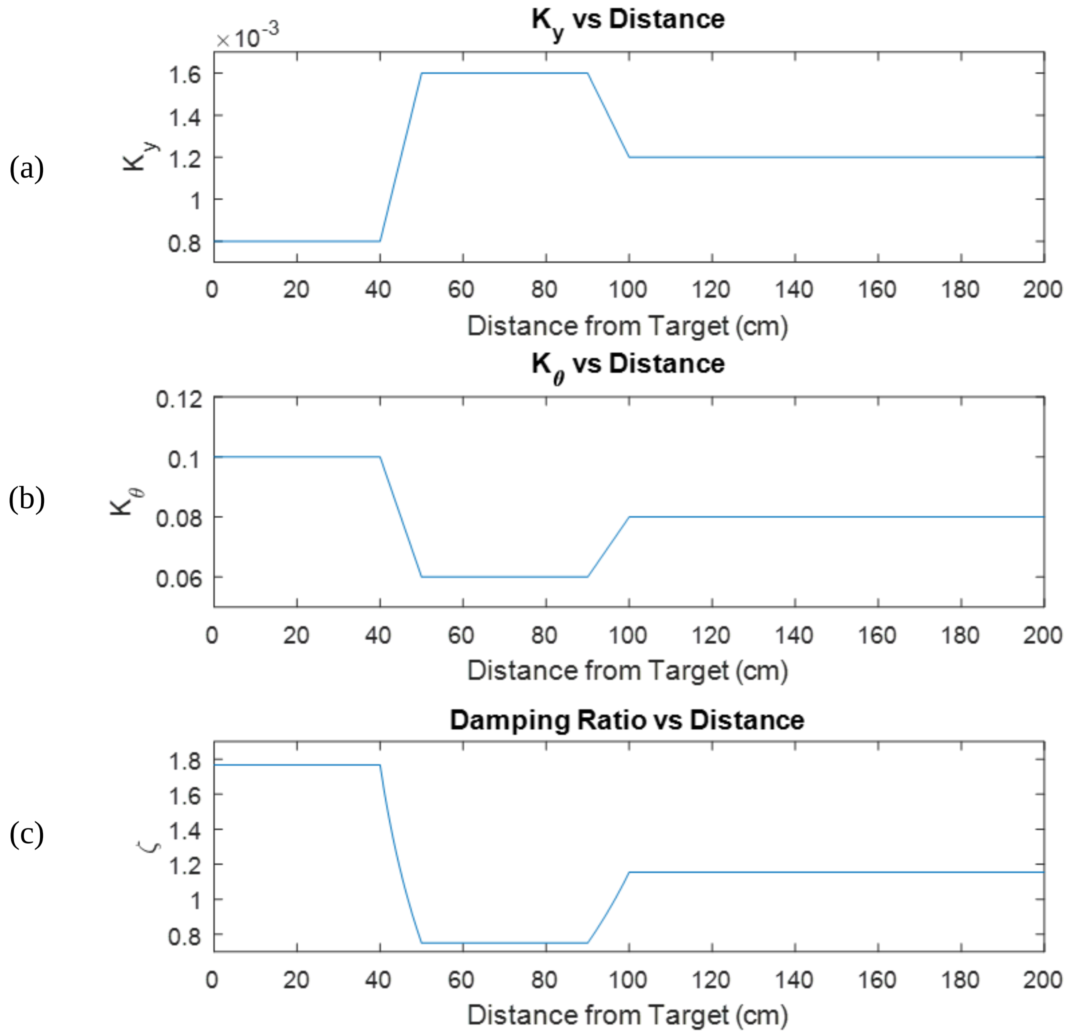


Figure 3.19: Control parameters used at different distances from the target. (a) K_y vs distance; (b) K_θ vs distance; (c) Theoretical damping ratio vs distance.

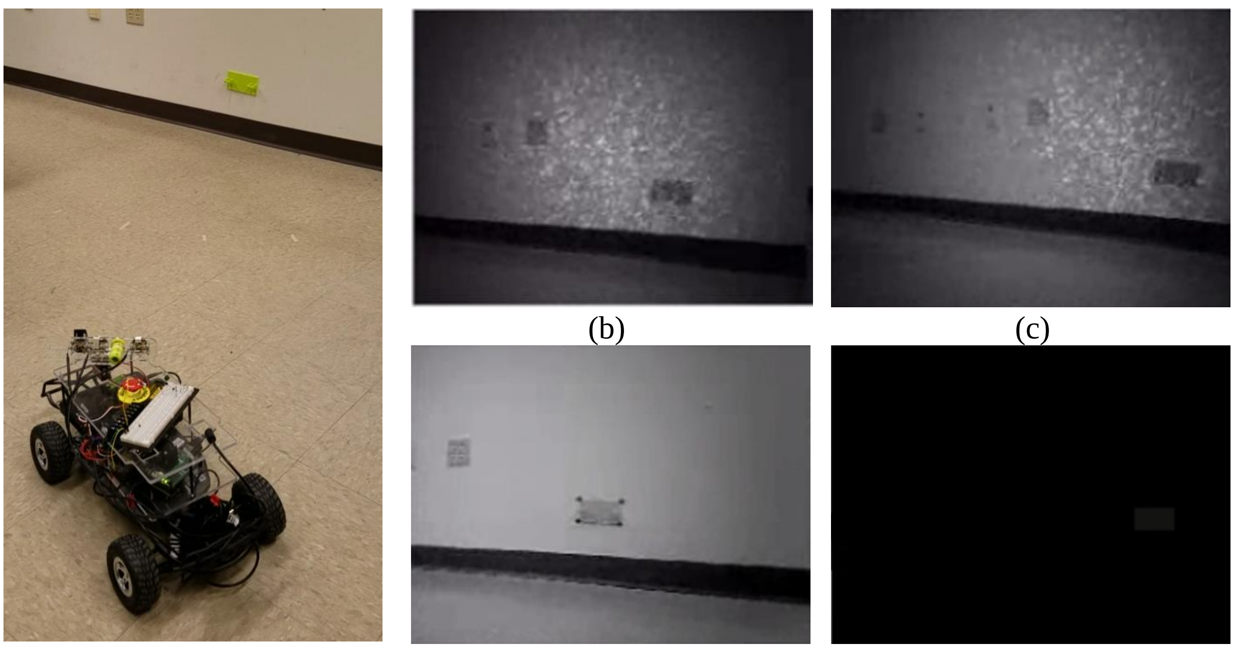
With this scheme incorporated, a test was completed with much better results. In this test, more data was collected, as six tests were completed at each of seven different starting angles of approach. As before, the vehicle started at a distance of 1.5 m from the target in these tests. The test results are given in Table 3.3, showing the vehicle starting angle from the target, the mean final measurement error for each angle of approach, and the standard deviation of that measurement error across the six tests at that angle of approach. Furthermore, based upon the mean results from each angle, a standard deviation is given, as well as a standard deviation across all 42 tests.

Table 3.3: Testing results using variable control parameters. The starting angle of approach is given for each test, and then the measurement mean and standard deviation across all tests for the specified angle of approach is given for each of x, y, and θ errors. Using the means from all the different angles of approach, the standard deviation of the error is calculated for each of x, y, and θ, and the standard deviation is also calculated for all 42 tests.

Starting Angle of Approach (°)	x error (mm) mean, std. dev.	y error (mm), mean, std. dev.	θ error (°), mean, std. dev.
-15	-0.8, 2.5	12.3, 1.5	2.1, 0.38
-10	0.0, 1.5	13.3, 2.4	1.4, 0.86
-5	-0.3, 1.6	10.0, 2.8	0.4, 0.38
0	1.0, 3.7	12.3, 2.4	0.1, 0.66
5	0.0, 2.3	12.2, 3.4	-0.6, 0.58
10	-1.3, 2.0	10.8, 4.0	-1.0, 1.10
15	-0.2, 1.2	8.3, 1.2	-2.0, 0.32
Std. dev. of means of each angle of approach	**0.7**	**1.7**	**1.4**
Std. dev. of all 42 tests	**2.2**	**3.0**	**1.5**

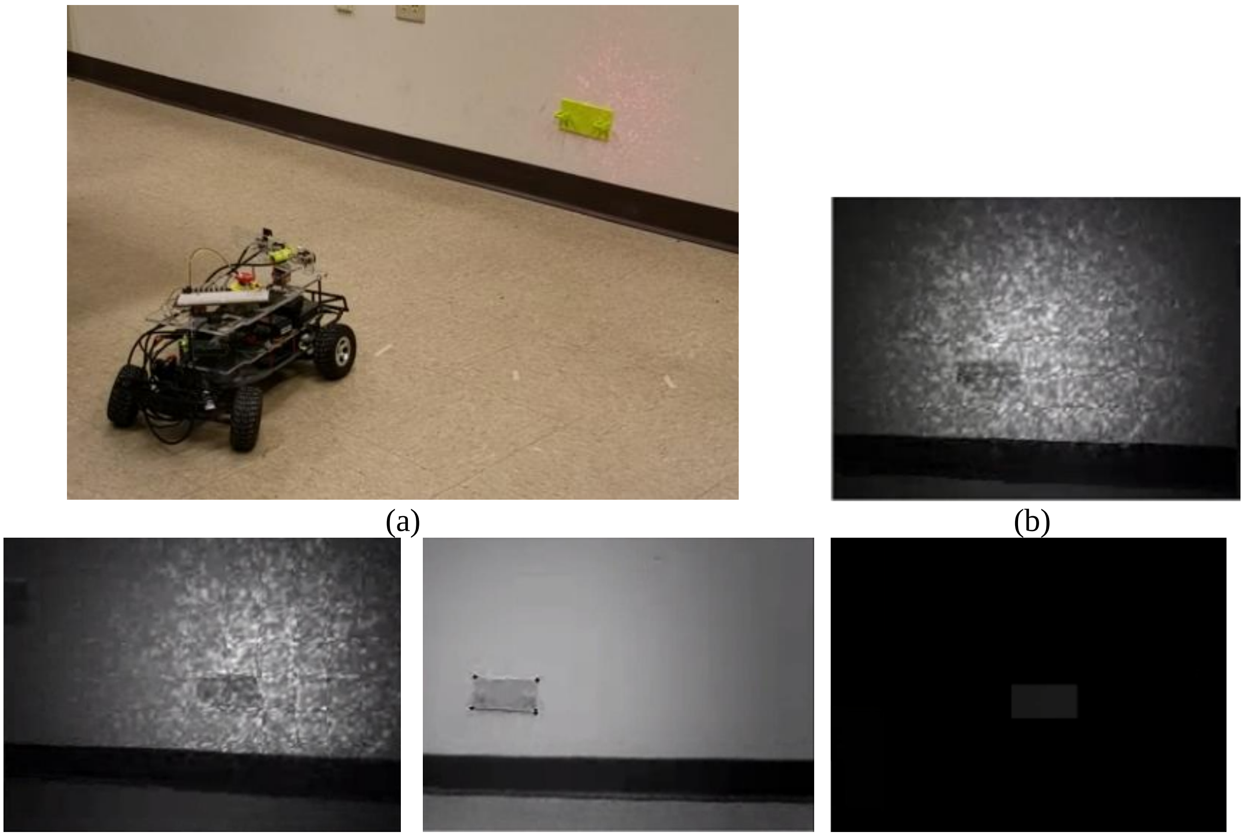
By evaluating the results given in Table 3.3, a couple of different types of error can be evaluated. First, the standard deviation for each angle of approach can be used to evaluate the uncertainty of a result, even when the vehicle starts from the same position. Laterally, the standard deviations ranged from 1.2 mm to 4.0 mm, while the standard deviation of the angle ranged from 0.38° to 1.10°. Another source of error is the error that is introduced by starting the vehicle from a more extreme starting position. From this, it was found that the standard deviation of the average lateral displacements from the different angles of approach that were tested was 1.7 mm, while the standard deviation of the angle across the different angles of approach was 1.4°, which characterizes the error which is due to the vehicle starting from a certain angle of approach. Overall for all the tests, the lateral standard deviation was found to be 3.0 mm and the angular standard deviation was found to be 1.5º, which characterizes the overall spread of the error, irrespective of the starting position.

In order to show the functionality of the system, a series of three figures are provided, which show the vehicle movement in a test that started 2.5 m away from the target and at a 30º angle to perpendicular from the target. The vehicle is shown at various stages of control, and the sensing during each stage is also shown. In each figure, (a) shows a picture of the vehicle and the target, (b)-(c) show the left and right images of the stereo camera, (d) shows the single camera texture image, and on this image the black dots show where the corners of the target are identified, and (e) shows the portion of the disparity map that is processed from the stereo camera. Figure 3.20 shows the system and processed data when the vehicle is in the region of the far distance control system, at more than 1.5 m from the target and outside of ±15º of perpendicular to the target. Figure 3.21 shows the system and processed data when the vehicle is between 1.15 m and 1.5 m from the target. Figure 3.22 shows the system and processed data when the vehicle is close to the target. For Figure 3.22 (e), recall from Section 3.11 that the edges of the target are not processed in the disparity map, in order to prevent portions of the target that are processed from leaving the field of view of the left or right camera at a close distance, which explains why the disparity map appears to have a short width in comparison with the target.



(a) (b) (c) (d) (e)

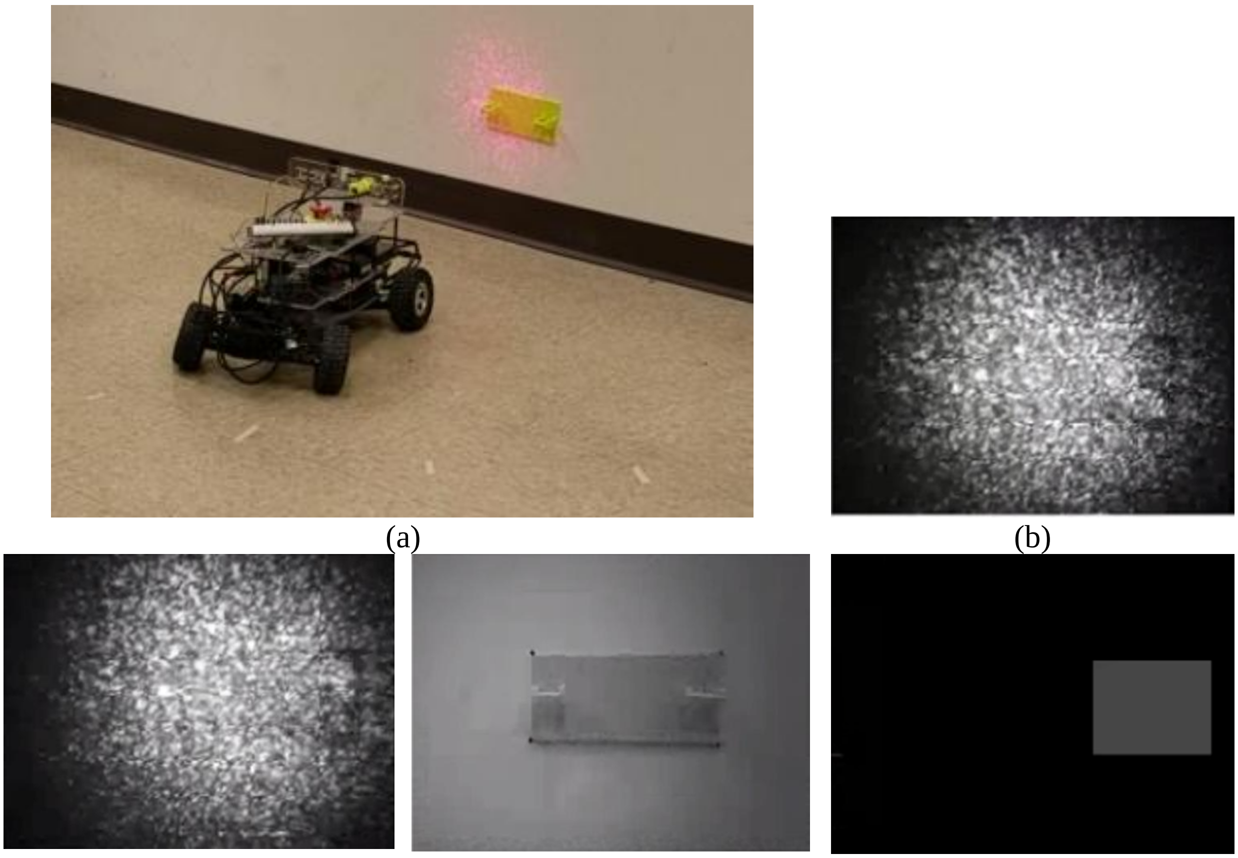
Figure 3.20: The vehicle moving towards the target using the far distance control system. (a) Photo of vehicle in motion; (b) Left image; (c) Right image; (d) Texture camera image, with black dots showing where the corners of the target were found; (e) Processed disparity map.



(a) (b)

(c) (d) (e)

Figure 3.21: The vehicle moving towards the target between 1.15 m and 1.5 m. (a) Photo of vehicle in motion; (b) Left image; (c) Right image; (d) Texture camera image, with black dots showing where the corners of the target were found; (e) Processed disparity map.



(a) (b)

(c) (d) (e)

Figure 3.22: The vehicle moving towards the target at a short distance. (a) Photo of vehicle in motion; (b) Left image; (c) Right image; (d) Texture camera image, with black dots showing where the corners of the target were found; (e) Processed disparity map.

4. CONCLUSIONS AND FUTURE WORK

In this paper, a sensor solution has been developed to solve the autonomous hitching problem, and the system has been tested on a model vehicle. The sensor is stereo vision based, and solves the stereo correspondence problem with a laser speckle projection system. According to an assessment of current literature and existing technology, this system is believed to be the first solution to the hitching problem with the following characteristics:

a) Sensor does not require modification of the hitch to sense the hitch
b) Sensor costs less than $1000
c) Sensor power requirement of less than 20 watts

The sensor exceeds these characteristics, as the total cost is $188 and the power requirement is 2.3 watts. Furthermore, the sensor is able to be used in close proximity with other sensors of the same type without loss of performance. In the testing of the vehicle, the system was shown to have an angular error of 1.5^{o}, with a lateral approach error of 3.0 mm, demonstrating sufficient accuracy to control the vehicle successfully to attach to a hitch. Furthermore, the system is able to work as long as the vehicle starts within 35^{o} of perpendicular to the hitch.

The work that has been completed has demonstrated that the sensor that has been developed is capable of being used as a sensor to solve the autonomous hitching problem. However, there are many areas of future work that can be explored in order to further improve and test the results.

First, further testing can be completed in other conditions, as the testing to this point has taken place in an indoor environment. For example, testing the system outdoors in sunlight could be an important step in verifying that the technology is able to be utilized in typical autonomous vehicle applications.

In this testing, it may also be worthwhile to explore implementing a different control scheme. Other, more recently-developed schemes exist that may be more suited for precise vehicle control. For example, an optimal control method such as a linear-quadratic regulator may be more suited to properly control the vehicle. This is certainly something worth investigation.

Additionally, the sensor could be used on the vehicle to detect obstacles near the target, and a path planning algorithm could be incorporated to move the vehicle to hitch while avoiding any obstacles in the path.

It may also be possible to further improve the speed of the sensor. Further restrictions to the block matching algorithm could be developed, or another, faster algorithm with comparable accuracy could be used. If this can be achieved, it may be possible to raise the stereo vision image resolution from 320×240 to 640×480. If this is able to be done, the system could see an additional improvement in sensor accuracy, which would also increase final vehicle positioning accuracy.

After additional tests are completed, the system can be used on a production vehicle, such as a tractor, for the function of attaching the tractor to a specific hitch. This will require some modification of the image processing diagrams. It may be advantageous to use a deep learning neural network to be able to identify the hitch from a single camera, and then retrain the neural network as new types of hitches are added to a library. This could be significantly more robust than the current color thresholding method of object identification. If these steps are taken, this technology could become the foundation of the first commercially available hitching system.

REFERENCES

[1] T. Ahamed, L. Tian, T. Takigawa and Y. Zhang, "Development of Auto-Hitching Navigation System for Farm Implements Using Laser Range Finder," *Transactions of the ASABE,* vol. 52, no. 5, pp. 1793-1803, 2009.

[2] V. Freese and B. Hendrickson, "Vehicle docking assistance system". United States of America Patent US8191915B2, 17 October 2008.

[3] D. Mattern, E. Tseng, J. Shaw, J. Plavcan, L. Yeung and T. Pilutti, "System to automate hitching a trailer". United States of America Patent US20180312022A1, 1 May 2017.

[4] R. Chaudhari, "Implement attachment guidance system". United States of America Patent US10351061B1, 9 March 2018.

[5] R. Wang, J. Peethambaran and D. Chen, "LiDAR Point Clouds to 3-D Urban Models: A Review," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* vol. 11, no. 2, pp. 606-627, 2018.

[6] R. Domínguez, E. Onieva, J. Alonso, J. Villagra and C. González, "LIDAR based perception solution for autonomous vehicles," in *2011 11th International Conference on Intelligent Systems Design and Applications,* Cordoba, Spain, 2011.

[7] A. Broggi, M. Buzzoni, S. Debattisti, P. Grisleri, M. Laghi, P. Medici and P. Versari, "Extensive Tests of Autonomous Driving Technologies," *IEEE Transactions on Intelligent Transportation Systems,* vol. 14, no. 3, pp. 1403-1415, 2013.

[8] J. Fan, X. Zhu and H. Yang, "Three-Dimensional Real-Time Object Perception Based on a 16-beam LiDAR for an Autonomous Driving Car," in *Proceedings of The 2018 IEEE International Conference on Real-time Computing and Robotics,* Kandima, Maldives, 2018.

[9] A. Davies, "This Lidar Is So Cheap It Could Make Self-Driving a Reality," WIRED, 7 July 2019. [Online]. Available: https://www.wired.com/story/lidar-cheap-make-self-driving-reality/.

[10] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough and A. Mouzakitis, "A Survey of the State-of-the-Art Localization Techniques and Their Potentials for Autonomous Vehicle Applications," *IEEE Internet of Things Journal,* vol. 5, no. 2, pp. 829 - 846, 2018.

[11] M. Hansard, S. Lee, O. Choi and R. Horaud, Time-of-Flight Cameras: Principles, Methods and Applications, London: Springer, 2013.

[12] S. Hsu, S. Acharya, A. Rafii and R. New, "Performance of a Time-of-Flight Range Camera for Intelligent Vehicle Safety Applications," in *Advanced Microsystems for Automotive Applications 2006*, Springer, Berlin, Heidelberg, 2006.

[13] S. Zhang, "High-speed 3D shape measurement with structured light methods: A review," *Optics and Lasers in Engineering,* vol. 106, pp. 119-131, 2018.

[14] P. Huang, C. Zhang and F. Chiang, "High-speed 3-D shape measurement based on digital fringe projection," *Optical Engineering,* vol. 42, no. 1, pp. 163-168, 2003.

[15] S. Zhang, "Recent progresses on real-time 3D shape measurement using digital fringe projection techniques," *Optics and Lasers in Engineering,* vol. 48, no. 2, pp. 149-158, 2010.

[16] N. Lazaros, G. Sirakoulis and A. Gasteratos, "Review of Stereo Vision Algorithms: From Software to Hardware," *International Journal of Optomechatronics,* vol. 2, no. 4, pp. 435-462, 2008.

[17] R. Hamzah and H. Ibrahim, "Literature Survey on Stereo Vision Disparity Map Algorithms," *Journal of Sensors,* vol. 2016, 2016.

[18] G. Bradski and A. Kaehler, Learning OpenCV, O'Reilly Media, Inc., 2008.

[19] A. Geiger, M. Roser and R. Urtasun, "Efficient Large-Scale Stereo Matching," in *Computer Vision – ACCV 2010*, 2010.

[20] X. Hu, Y. Wu, D. Zhang, L. Qian and L. Wu, "Research on Improvement of Stereo Matching Algorithm," *Journal of Computers,* vol. 19, no. 4, pp. 110-121, 2018.

[21] M. Brown, D. Burschka and G. Hager, "Advances in Computational Stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 25, pp. 993-1008, 2003.

[22] J. Diebel, K. Reuterswad, S. Thrun, J. Davis and R. Gupta, "Simultaneous Localization and Mapping with Active Stereo Vision," in *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems* , Sendal, Japan, 2004.

[23] A. Koschan, V. Rodehorst and K. Spiller, "Color Stereo Vision Using Hierarchical Block Matching," in *Proceedings of 13th International Conference on Pattern Recognition*, Vienna, Austria, 1996.

[24] W. Jang, C. Je, Y. Seo and S. Lee, "Structured-light stereo: Comparative analysis and integration of structured-light and active stereo for measuring dynamic shape," *Optics and Lasers in Engineering,* vol. 51, no. 11, pp. 1255-1264, 2013.

[25] M. Schaffer, M. Grosse and R. Kowarschik, "High-speed pattern projection for three-dimensional," *Applied Optics,* vol. 49, no. 18, pp. 3622-3629, 2010.

[26] F. Zhong, R. Kumar and C. Quan, "RGB laser speckles based 3D profilometry," *Applied Physics Letters,* vol. 114, no. 20, 2019.

[27] D. Khan, M. Shirazi and M. Kim, "Single shot laser speckle based 3D acquisition system for medical applications," *Optics and Lasers in Engineering,* vol. 105, pp. 43-53, 2018.

[28] J. Dainty, Laser Speckle and Related Phenomena, New York: Springer-Verlag Berlin Heidelberg, 1975.

[29] S. Zhang, High-speed 3D imaging with digital fringe projection techniques, CRC Press, Taylor & Francis Group LLC, 2016.

[30] C. McCormick, "Stereo Vision Tutorial - Part I," January 2014. [Online]. Available: http://mccormickml.com/2014/01/10/stereo-vision-tutorial-part-i/. [Accessed August 2019].

[31] J. Hyun and S. Zhang, "High-speed 3D imaging with three binary patterns using Hilbert transform," in *Proceedings Volume 11102, Applied Optical Metrology III*, San Diego, 2019.

[32] J. Laumond, S. Sekhavat and F. Lamiraux, Guidelines in Nonholonomic Motion Planning for Mobile Robots, Springer, Berlin, Heidelberg, 1998.

[33] Y. Kanayama, Y. Kimura, F. Miyazaki and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *Proceedings., IEEE International Conference on Robotics and Automation*, Cincinnati, 1990.

[34] S. Suzuki and K. Abe, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing,* vol. 30, no. 1, pp. 32-46, 1985.

[35] D. Douglas and T. Peucker, "Algorithms for the reductions of the number of points required to represent a digitised line or caricature," *The Canadian Cartographer,* vol. 10, no. 2, pp. 112-122, 1974.

[36] V. Lepetit, F. Moreno-Noguer and P. Fua, "EPnP: An Accurate O(n) Solution to the PnP Problem," *International Journal of Computer Vision*, vol. 81, 2009.

[37] H. Kim, S. Leutenegger and A. Davison, "Real-Time 3D Reconstruction and 6-DoF Tracking with an Event Camera," in *Computer Vision – ECCV 2016*, 2016.